The University of Jordan

Authorization Form

I, Elham Fahmey Ahmad, authorize the University of Jordan to supply copies of my Thesis/ Dissertation to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature:

Date: 27/4/2011

الجامعة الأردنيـــــة
كلية الدراسات العليا

التاريـــخ: ٢٧ / ٤ / ٢٠١١

نموذج رقم (١٨)
اقرار والتزام بالمعايير الأخلاقية والأمانة العلمية
وقوانين الجامعة الأردنية وأنظمتها وتعليماتها
لطلبة الماجستير

أنا الطالب: الهام فخري موسى أحمد الرقم الجامعي: (8080206)
تخصص: علم الحاسوب الكليـــــة: الملك عبدالله الثاني لتكنولوجيا المعلومات

عنوان الرسالة: Gynecological Diseases Diagnosis Using Clinical Decision Support System

اعلن بأنني قد التزمت بقوانين الجامعة الأردنية وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة باعداد رسائل الماجستير عندما قمت شخصيا" باعداد رسالتي وذلك بما ينسجم مع الأمانة العلمية وكافة المعايير الأخلاقية المتعارف عليها في كتابة الرسائل العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيسا" على ما تقدم فانني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في الجامعة الأردنية بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي أي حق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب: .... التاريخ: ٢٧ / ٤ / ٢٠١١

GYNECOLOGICAL DISEASES DIAGNOSIS USING CLINICAL DECISION SUPPORT SYSTEM

By
Elham Fahmey Ahmad

Supervisor
Dr. Mohammad Alshraideh

Co- Supervisor
Dr. Kamil Fram

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree of Science in Computer Science

Faculty of Graduate Studies
The University of Jordan

April, 2011

COMMITTEE DECISION

This Thesis /Dissertation (Gynecological Diseases Diagnosis Using Clinical Decision Support System) was successfully Defended and Approved on 13/4/2011.

Examination Committee	Signature
Dr. Mohammad Alshraideh (Supervisor) Assistant Professor of Software Testing Using Evolutionary Algorithms	
Dr. Kamil Fram (Co-Supervisor) Associate Professor of Medicine	
Dr. Abdel latif Abu-Dalhoum (Member) Associate Professor of Evolutionary Algorithms and DNA Computation Models	
Dr. Azzam Sleit (Member) Associate Professor of Algorithms and Imaging Data Bases	
Dr. Ahmad Khasawneh (Member) Assistant Professor of Computer Information Systems (The Hashemite University)	

DEDICATION

I dedicate the success of this work…

To My Lord Allah, with endless thanks, praise be to God for all his graces, and for his greatest gift (The Islam)

To the Seal of the Prophets and his nation the beloved Mohammad (God's blessing and peace be upon him)

To My dearest husband Yahya Suliman and to my beloved son Elyas, for their immense patience, support, and encouragement throughout the duration of this thesis, words cannot adequately express how vital their support has been to the completion of this thesis.

شكر وعرفان

اللهم لك الحمد كما ينبغي لجلال وجهك و عظيم سلطانك

الحمد الله عز وجل على النعم التى لا تعد و لا تحصى و أعظمها نعمة الإسلام و اللهم صلى على أشرف المرسلين وخاتم النبيين سيدنا و حبيبنا محمد (صلى الله عليه وسلم).

وأتوجه بالشكر و الامتنان إلى زوجي العزيز يحيى سليمان و ابني الحبيب إلياس ، حيث أن الكلمات تقف عاجزة عن التعبير عن مدى دعمهما و صبرهم طوال مدة إنجاز هذه الرسالة.

ACKNOWLEDGEMENT

The name of Allah, the Most Beneficent Most Merciful

All praise is due to **Allah**, the source of all knowledge and strength. I acknowledge His infinite mercy and grace in making this work a success. May His peace and blessings be upon His final messenger **Muhammad**, a guidance and inspiration to our lives.

The successful completion of this work was made possible by a number of major contributions from different persons, to whom I wish to express my due gratitude.

I would like to thank:

Dr. Muhammad Al-Shraideh, thesis supervisor, for his extraordinary supervision and invaluable which made this work possible, for his excellent guidances and helpful advices. Whenever I lost sight of the objectives of this thesis, he skillfully led me back on track. In addition to his humility and great style are dealing and respect for others.

Dr. Basel Mahafzah for teaching methods of scientific research and providing guidance and helpful advices in all subjects throughout the duration of this thesis.

Dr. Mousa Al-Akhras for providing fresh ideas and directions in both areas of machine learning and neural networks.

Dr. Kamil Fram for welcoming me in his clinic and his manner gentle and convivial .I would also like to thank his resident **Dr. Mohammad Ramadneh** for helping me understand medical issues optimally. With special thanks for all patients, for helping me.

Special thanks and sincere gratitude to Yahya Suleiman and Elyas, my dearest husband and my beloved son, for their immense patience, support, and encouragement throughout

the duration of this thesis. Words cannot adequately express how vital their support has been to the completion of this thesis.

To all my friends, in particular, Nesreen Hamad, Tasneem Abu-Kabeer, Heba Saadeh, and others thanks all for their support and encouragement. Without them, I won't be able to get out from the bottle neck easily.

Finally, I am so grateful to **Mr. Mohammad Albatayneh** and **Ms. Soa'd Rashed** for their efforts to accomplish this task optimally, by helping to extract the files required in collecting data and lists of patients, for helping me and understanding my needs.

Elham F. Ahmad

Amman 2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	***Meaning***
Abor	Abortion
Amen	Amenorrhea
ANN	Artificial Neural Network.
ANNs	Artificial Neural Networks.
CDSS	Clinical Decision Support System.
CDSSs	Clinical Decision Support Systems.
CI	Computational Intelligence.
Dys	Dysmenorrhea
Fib	Fibroids.
GRNN	General Regression Neural Network.
Hardlim	hard-limit transfer function
Hyper	Endometrial Hyperplasia
Inc	Infection.
Inf	Infertility
JUH	Jordan University Hospital.
LF	Linear Function.
LM	Levenberg–Marquardt.
Mg	Migrants
MLP	Multilayer Perceptron.
MLPs	Multilayer Perceptrons.

NN	Neural Network.
PID	Pima India Diabetes
PNN	Probabilistic Neural Network.
Pol	Polyps.
Pro	Prolapse
RBF	Radial Basis Network.
Rprop	Resilient Back Propagation Algorithm.
SOM	Self Organizing Maps.
TANH	Hyperbolic Tangent Sigmoid Function.
UCI	University of California Irvine
WBCD	Wisconsin Breast Cancer Data set.

GYNECOLOGICAL DISEASES DIAGNOSIS USING CLINICAL DECISION SUPPORT SYSTEM

By
Elham Fahmey Ahmad

Supervisor
Dr. Mohammad Alshraideh

Co- Supervisor
Dr. Kamil Fram, MD

ABSTRACT

Gynecological disease diagnosis is one of the important issues in the medical field world wide and the problem is also faces in Jordan. Gynecologist has to analyze and diagnose the disease according to symptoms are various and similar. Also the gynecologist may accidentally miss out some symptoms. Hence, there is a need for developing a decision support system in order to assist gynecologist in the diagnosing process.

In the last decades, several tools and various methodologies have been proposed by the researchers for developing effective Clinical Decision Support Systems (CDSSs) in order to improve the ability of the physicians to do their work, one of these methodologies is Artificial Neural Networks (ANN) that are computer paradigms that belong to the Computational Intelligence (CI) family and are inspired by the biological neural system.

In this thesis, a Multilayer Perceptron (MLP) Feed Forward Neural Network is used to develop a Clinical Decision Support System (CDSS) for the diagnosis of eleven gynecological diseases. The input layer of the system includes 54 input variables categorized into eleven groups. And use encoding the proposed coding schemes. Whereas, the output layer contains one neuron that will produce a number that represents the gynecological disease type based on patient case. An iterative process used to determine the number of layers and nodes in the hidden layers. Furthermore, a Resilient back propagation algorithm (Rprop) is used to train the system. In particular, a 10-fold cross validation scheme was used to access the generalization of the proposed system. The experiments were performed with the proposed system. We obtained 94.5% classification accuracy from the experiments made on data taken from 550 patients' medical records suffering from eleven gynecological diseases managed at the gynecological clinics at the Jordan University Hospital (JUH).

Chapter One

1. Introduction

A major class of problems in the medical science involves the diagnosis of disease, based upon various symptoms and tests performed upon the patient. When several symptoms and tests are involved, the ultimate diagnoses may be difficult to obtain, even for a medical expert. Furthermore, similar symptoms for multiple diseases may boost the chance for a medical misdiagnosis. Consequently over the past few decades, this has given rise to computerized diagnostic tools, proposed to assist the physician to clarify the confusing data in case of imprecision and uncertainty (Kiyana and Yildirim, 2003). These computerized tools are called Clinical Decision Support Systems **(CDSSs)** which are types of computer programs that can be developed through several kinds of Machine Learning techniques, and are used to assist clinicians at the point of care (Yan et al., 2006).

In this chapter, section 1.1 illustrates a general overview of machine learning, section 1.2 concerns about CDSSs, section 1.3 gives a general overview of gynecological diseases, section 1.4 introduces the problem statement and thesis motivation and thesis organization is illustrated in section 1.5.

1.1 Machine Learning Overview

Machine Learning studies computer algorithms for learning to do stuff, learning to complete a task, or to make accurate predictions, or to behave intelligently. The learning done is always based on some sort of observations or data, such as examples (the case in this thesis), direct experience or instruction. Hence, in general machine learning is about

learning to do better in the future based on what was experienced in the past (Mitchell, 1997).

Learning, like intelligence, covers such a broad range of processes that it is difficult to define precisely. A dictionary definition includes phrases such as "to gain knowledge, or understanding of, or skill in, by study, instruction, or experience," and "modification of a behavioral tendency by experience." Zoologists and psychologists study learning in animals and humans (Mitchell, 1997). There are several parallels between animal and machine learning. Certainly, many techniques in machine learning derive from the efforts of psychologists to make more precise their theories of animal and human learning through computational models. It seems likely also that the concepts and techniques being explored by researchers in machine learning may illuminate certain aspects of biological learning (Fausett, 2001).

As regards machines, we might say, very broadly, that a machine learns whenever it changes its structure, program, or data (based on its inputs or in response to external information) in such a manner that it's expected future performance improves. Some of these changes, such as the addition of a record to a data base, fall comfortably within the province of other disciplines and are not necessarily better understood for being called learning. But, for example, when the performance of a speech-recognition machine improves after hearing several samples of a person's speech, we feel quite justified in that case saying that the machine has learned(Veropoulos, 2001).

One might ask "Why should machines have to learn?" There are several reasons why machine learning is important. Of course, we have already mentioned that the achievement

of learning in machines might help us understand how animals and humans learn. But there are important engineering reasons as well (Mitchell, 1997). Some of these are:

- Some tasks cannot be defined well except by example; that is, we might be able to specify input/output pairs but not a concise relationship between inputs and desired outputs. We would like machines to be able to adjust their internal structure to produce correct outputs for a large number of sample inputs and thus suitably constrain their input/output function to approximate the relationship implicit in the examples (Fausett, 2001).

- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans. Machines that learn this knowledge gradually might be able to capture more of it than humans would want to write down.

- Environments change over time. Machines that can adapt to a changing environment would reduce the need for constant redesign.

- New knowledge about tasks is constantly being discovered by humans. Continuing redesign of AI systems to conform to new knowledge is impractical, but machine learning methods might be able to track much of it.

The Machine Learning techniques have been employed by many disciplines to automate complex decision making and problem solving tasks. This field of research is immensely diverse, with applications in medicine, mathematics, computer science,

chemistry, economics, business management and many other fields (Ramalingam et al., 2006).

The majority of these techniques are concerned in pattern recognition problems. Pattern recognition is the act of taking in raw data and taking an action based on the "category" or the pattern (Duda et al., 2001). An example of pattern recognition problems is classification (Mitchell, 1997). In machine learning, the solution of pattern recognition problems lies within the field of supervised learning (the base for this thesis). The task is to learn (induce) the relationship between the dependent attributes (input) and the designated attribute (output) from a set of examples, i.e. generalize information collected from given data to unseen data. A common supervised machine learning tools are: Multilayer Perceptron (MLP) neural network (Yan et al., 2006) and decision trees (Mitchell, 1997). Another category of problems in machine learning is unsupervised learning .unsupervised learning is a class of problems in which one seeks to determine how the data are organized. It is distinguished from supervised learning in that the learner is given only unlabeled examples (Geoffrey and Terrence,1999). One form of unsupervised learning is clustering which is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense among neural network models. The Self-organizing map **(SOM)** is an example of unsupervised machine learning tool (Geoffrey and Terrence,1999).

1.1.1 Application of Machine Learning

Because of their ability to identify patterns or trends in data, machine learning methods have been successfully applied in many industries review of (Fausett, 2001):

- **Face recognition**: it is one of many security application, that the system classify the faces into one if many categories in the other words, it recognize who this person is.

- **Spoken language** understanding: within the context of a limited domain, determine the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories.
- **Spam filtering**: identify email messages as spam or non-spam.
- **Credit card fraud detection**: identify credit card transactions (for instance) which may be fraudulent in nature.
- **Weather prediction**: predict, for instance, whether or not it will rain tomorrow.
- **Medical diagnosis**: diagnose a patient as a sufferer or non-sufferer of some diseases.

1.1.2 Machine Learning in Medical Domain

Machine Learning algorithms were from the very beginning designed and used to analyze medical data sets. Today machine learning provides several indispensable tools for intelligent data analysis. Especially in the last few years, the digital revolution provided relatively inexpensive and available means to collect and store the data. Modern hospitals are well equipped with monitoring and other data collection devices, and data is gathered and shared in large information systems (Veropoulos, 2001). Machine learning technology

is currently well suited for analyzing medical data, and in particular there is a lot of work done in medical diagnosis in small specialized diagnostic problems. Data about correct diagnoses are often available in the form of medical records in specialized hospitals or their departments. All that has to be done is to input the patient records with known correct diagnosis into a computer program to run a learning algorithm (Fausett, 2001). This is of course an oversimplification, but in principle, the medical diagnostic knowledge can be automatically derived from the description of cases solved in the past. The derived classifier (usually called CDSSs) can then be used either to assist the physician when diagnosing new patients in order to improve the diagnostic speed, accuracy and/or reliability, or to train students or physicians non-specialists to diagnose patients in a special diagnostic problem (Mitchell, 1997).

1.2 Clinical Decision Support Systems Overview

In medical diagnosis, some of the scopes for ambiguity in the inputs are the history (patient’s description of the disease condition), physical examinations (especially in cases of uncooperative patients) and laboratory tests (faulty methods or equipment). Moreover, in the treatment there are chances of drug reactions and specific allergies (Brause, 2001). All these factors may not be taken into account by a busy clinician attending hundreds of patients daily. Therefore, computers can help the clinician to reach an accurate diagnosis faster because they make fewer errors than humans in making analyses of complex data (Rasheed, 2009).

In the literature, many researchers have given their definitions of CDSSs. Musen (1997) defined a CDSS as any piece of software that takes information about a clinical situation as inputs and that produces inferences as outputs that can assist practitioners in their decision making and that would be judged as "intelligent" by the program's users. Miller and Geissbuhler (2007) defined a CDSSs providing diagnostic decision support as a 'computer-based algorithm that assists a clinician with one or more component steps of the diagnostic process. Sim as cited in (kong et al., 2008) defined CDSSs as '' software that was designed to be a direct aid to clinical decision-making, in which the characteristics of an individual patient are matched to a computerized clinical knowledge base and patient specific assessments or recommendations are then presented to the clinician or the patient for a decision''. So in general CDSSs are computer programs designed to help healthcare professionals to make clinical decisions. In a sense, any computer system that deals with clinical data or knowledge is intended to provide decision support .These systems can be used for diagnosis, treatment of health diseases, and future evaluation of the patient (Kong et al., 2008).

The implementation of these systems could have several beneficial effects; they improved patient safety by for example through reducing medication errors and adverse events, medical errors implies actions such as the conclusion of wrong diagnosis ,the prescription of an incorrect treatment or the inefficient execution of correct treatment. Additionally, CDSSs improved clinical documentation and patient satisfaction. Moreover, they supported medical education and training and provided immediate feedback to patients (Thornett, 2001). Because of these benefits, CDSSs were applied in many medical

domains, such as cardiology, gynecology, diabetes, etc (Veropoulos, 2001). In this thesis we will focus on the gynecology domain.

1.3 Gynecological Diseases

Gynecology word comes from the Greek, whereas gynaika (γυναίκα) meaning woman is the medical practice dealing with the health of the female reproductive system (uterus, vagina and ovaries). Literally, outside medicine, it means "the science of women". It is the counterpart to andrology, which deals with medical issues specific to the male reproductive system. The female reproductive tract is a complex and intricate system. It must stay balanced in order to remain healthy (Edozien, 2007).

As every woman knows, there could be myriad gynecological conditions. Most of the times, women tend to ignore symptoms of gynecological diseases or any kind of infection (Penny and Foy, 2007).They are typically too embarrassed to discuss gynecological symptoms with their doctors, or are afraid to approach a qualified gynecologist. Just to being with, most gynecological problems relate to different varieties of infections. For instance, vaginal infections are extremely frequent occurrences across the globe. Typically, it is a yeast infection which is rather common place. Such an infection is nothing to be ashamed of, as it does not indicate anything conclusive. Almost all women have had to deal with such infections which may result because of organisms which would generally be present, even in a vagina that is perfectly healthy. As there are countless gynecological diseases and variations of the same, one has to understand that gynecological problems will have multiple and varied symptoms (Chard and Schreiner, 1990).

There are many diseases that woman can be faced, those diseases are multiple and different their symptoms, varies and similar among themselves, as result to the large number of those diseases, can not be restricted and assembled in one research. And may depend upon an increase or decrease in the spread of any disease of these diseases and on the place or the environment where patient live (Chard and Schreiner, 1990).

In particular, there are varies diseases that spread in Jordan, exhibiting the most important diseases and that are frequently diagnosed in the gynecologist's clinics. And this thesis include these diseases like: Polyps(Cervical, Endometrial), Infections(Cervicitis, Endometritis, Cervical Erosion, Endometrial atrophy), Fibroids, Prolapse, Cancer(Cervical ,Uterine), Endometrial hyperplasia, Migrants(Adenomyosis, Endometriosis), Amenorrhea, Abortion , Dysmenorrhea and Infertility. Discussing these diseases briefly:

- *Polyps*: Polyp is growth or protrusion that extends from a mucous membrane. The polyps divide into two main categories are; cervical polyps and endometrial (Uterus) polyps, endocervical, cervical and endometrial polyps are the most common benign neoplostic growth of the cervix and uterus polyps are smooth, soft, reddish purple to cherry red and fragile like growth projections that hangs from cervix of the women uterus. The cervix is the organ that provides an opening between the vagina and the uterus, while the endometrial polyps are an overgrowth of tissue in the endometrium (Penny and Foy, 2007).

- *Infections*: Infections that choice in this research includes; Cervicitis, Endometritis, Cervical Erosion, and Endometrial atrophy. Cervicitis is Inflammation of the cervix

while Endometritis is Inflammation of the endometrium (uterine lining). But Cervical erosion: A shallow superficial ulceration of the cervix while Endometrial atrophy is Atrophy in the cells of the uterus (Edozien, 2007).

- *Fibroids*: The third disease is Fibroids; uterine fibroids are nodules of smooth muscle. They are benign tumors of muscle cell origin .they are the most common tumor in woman (Penny and Foy, 2007).

- *Prolapse:* Prolapse is protrusion or Prolapse of the uterus into the wrong position. They can roughly be separated into three categories as following :

  a. Uterine Prolapse: Protrusion of the cervix and uterus into the barrel of the vagina.

  b. Cystocele: Protrusion of the bladder into the vagina, signifying the relaxation of fascial supports of the bladder.

  c. Rectocele: Protrusion of the rectum into the vagina, signifying a relaxation of rectal supports.

- *Cancer:* cancer is a class of diseases in which a group of cells display uncontrolled growth, invasion that intrudes upon and destroys adjacent tissues, and sometimes metastasis, or spreading to other locations in the body. In gynecology , there are many types of cancer diseases that two types which only will be discussed are ; cervical cancer and uterine, In which are the most common two cancers of the

female reproductive system, specifically the cervix of the uterus. Cervical cancer can be fatal if left untreated, while uterine cancer is a malignant disease that occurs when there is an uncontrolled growth of cancerous cells (abnormal cells) in the uterus.

- *Endometrial hyperplasia*: A proliferation of the endometrial cells resulting in glandular enlargement and budding. The proliferation may or may not be associated with atypia of the endometrial cells. When the hyperplastic changes are excessive, there is formation of complex epithelial structures (complex endometrial hyperplasia), that may be progressed to frank but most likely clinically malignant condition in the future (Penny and Foy, 2007).

- *Migrants*: Migrants include; two different diseases, Adenomyosis means growth of the endometrial glands and stroma into the uterine myometrium (muscle layer of the uterus) to a depth of at least 2.5 mm from the tasalis layer of the endometrium. It is one of the most common gynecological diseases effecting 30%-50% of women at the age of fifty years. The condition is generally harmless but can be very painful. While Endometriosis, is a common disorder of a woman's reproductive organs and pelvic area. Endometriosis is often associated with painful menstrual periods, and if left untreated, can result in complications, such as infertility and the formation of painful adhesions. Endometriosis is not curable in many cases, but it can be successfully treated to minimize discomfort and increase the chances of conceiving a pregnancy. Endometriosis is benign, progressive disease; mean the presence and growth of glands and stroma identical to the lining of the uterus in an aberrant

location. It has special qualities because it responds to changing hormones by becoming thickened and engorged with blood during each menstrual cycle (Penny and Foy, 2007).

- *Amenorrhea*: is the absence of a menstrual period in a woman during the reproductive age. Amenorrhea is a symptom with many potential causes. Primary amenorrhea (menstruation cycles never starting) may be caused by developmental problems such as the congenital absence of the uterus, or failure of the ovary to receive or maintain egg cells. Also, delay in pubertal development will lead to primary amenorrhea. It is defined as an absence of secondary sexual characteristics by age 14 with no menarche or normal secondary sexual characteristics but no menarche by 16 years of age. Secondary amenorrhoea (menstruation cycles ceasing) it can be either physiologic (pregnancy) or pathologic which is often caused by hormonal disturbances from the hypothalamus and the pituitary gland, from premature menopause (Penny and Foy, 2007).

- *Abortion:* Termination of a pregnancy before 20 weeks gestation calculated from date of onset of last menses. An abortion can occur spontaneously due to complications during pregnancy or can be induced in humans and other species (Edozien, 2007).
- *Dysmenorrhea:* Painful menstruation with a cramping sensation in the lower abdomen or discomfort due to menstruation.

- *Infertility*: Failure of conception after one year of marriage in spite of having regular, normal, frequent unprotected sexual intercourse(Edozien, 2007).

Each disease of those diseases has many different symptoms that shown in **Table 1.1** but must remember that not all of the symptoms can exist when being a patient and it depends on extent status of the patient and severity of her disease.

Table 1.1: all diseases and their symptoms

No	Diseases	Symptoms and signs
1	Polyps	• Intermenstrual bleeding. • Postmenopausal bleeding. • Excessive bleeding during a menstrual period(menorrhagia) • Spotting bleeding after intercourse. • Abnormal Vaginal discharge
2	Infection	• Spotting bleeding after/during intercourse • Abnormal Vaginal bleeding • Intermenstrual bleeding. • scanty menstrual blood (spotting) • Dysmenorrheal • Lower abdominal pain • Fever • Abnormal Vaginal discharge which may be : ○ Non-smelly vaginal discharge ○ Smelly vaginal discharge ○ Yellow vaginal discharge ○ Vaginal discharge containing blood
3	Fibroids	• Painful bleeding during menstruation. • Frequent urination. • Pain during sexual intercourse. • Bleeding between periods (heavy irregular bleeding). • Low back pain, pelvic pain • Painful urination
4	Prolapse	• Urinary frequency • Urinary incontinence • Difficulty stopping urination • Difficulty starting urination • Dragging sensation in lower abdomen • Difficult or painful urination • Feeling of rectal fullness • A feeling of fullness in the lower abdomen • Increased urge to urinate • Painful bowel with cycles of diarrhea and constipation. • Difficulty emptying the bowel • Valur pain
5	Cancer	• Abnormal vaginal bleeding • Difficult or painful urination

		• Post coital bleeding • Abnormal vaginal bleeding • Vaginal discharge containing blood • Painful sex (Dysparunia) • Vaginal discharge • Pelvic pain • Loin pain • Decrease appetite
6	Endometrial hyperplasia	• Heavy menstruation • Extended menstruation • Menorrhagia • Irregular menstruation
7	Migrants	• Several menstrual cramps. • Painful intercourse. • Painful orgasms. • Painful bowel movements often with cycles of diarrhea and constipation. • low back pain, pelvic pain • Pain before and/or after menstruation. • Heavy or irregular menstrual bleeding. • Intestinal distress such as bloating, vomiting, nausea. • Painful menstruation • Longer periods
8	Amenorrhea	• Increase appetite or decrease appetite. • Lower abdominal pain • Painful intercourse. • Redundant breast • Dry skin. • Abnormal hair growth mainly at face
9	Abortion	• Vaginal bleeding • Uterine bleeding • Menstrual cramps • Lower abdominal pain • Trauma
10	Dysmenorrhea	• Painful menstruation • Menstrual cramping
11	Infertility	• Genital infection • Pelvic infection • Endocrine dysfunction • Sexual dysfunction • Male factor

1.4 Problem Statement and Thesis Motivation

Brause (2001) showed in his study that machine learning can help in making correct diagnoses by reducing medical errors where the number of deaths caused by medical error has been estimated to be over 98.000 per year. Having an effective CDSS will significantly reduce this number.

Gynecologist has to analyze and diagnose the disease according to many and similar symptoms. The patient may not be in apposition to spell out all her problems or may not know about symptoms of a disease and even the disease itself, and the gynecologist may accidentally miss out some the symptoms. All gynecologists in the world may not have knowledge about all types of gynecological diseases. Hence, there is a need for developing a decision support system in order to assist gynecologist in the diagnosing process.

From these results this thesis aims to develop a CDSS based on the patients' clinical records that were collected from the gynecology clinics at Jordan University Hospital (JUH) for eleven types of gynecological diseases; Polyps, Infection, Fibroids, Prolapse, Cancer, Endometrial hyperplasia, Migrants, Amenorrhea, Abortion, Dysmenorrhea and Infertility. In order to accomplish this, a neural network with a resilient back propagation will be used. This thesis will also investigate the number of hidden neurons and hidden layers to observe its impact on the classification accuracy.

1.5 Thesis Organization

This thesis is organized as follows: preliminaries concepts and background information related to ANN and some literature review are presented in Chapter 2. Data preparation and system architecture are illustrated in Chapter 3. The experimental results of different architectures that were applied on the collected data sets are presented in Chapter 4. Chapter 5 concludes this research and proposes some future works.

Chapter Two

2. Literature Review

2.1. Artificial Neural Networks Background

Computational Intelligence **(CI)** is a set of nature-inspired computational methodologies and approaches to address complex problems of the real world applications to which traditional (first principals, probabilistic, etc.) methodologies and approaches are ineffective or infeasible (Thammano and Moolwong, 2009). One of the techniques that belong to the CI family is Artificial Neural Networks **(ANNs)** (usually called Neural Network **(NN)**) that were introduced in 1940s (further developed in 1980s) as an attempt to emulate the features of biological neural networks in order to address a range of difficult information processing, analysis and modeling problems (Fausett, 2001). They are a powerful data modeling tool that is able to capture and represent complex input/output relationships. Since the motivation for the development of NN technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain; as a result neural networks resemble the human brain in the following two ways:

1. A neural network acquires knowledge through learning (experience).

2. A neural network's knowledge is stored within inter-neuron connection strengths known as weights.

ANNs have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse

as finance, medicine, engineering, geology and physics (Mehrotra et al., 1996). Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced (Haykin, 2001). Which its success can be attributed to a few key factors:

- **Power:** Neural network models can implicitly detect complex nonlinear (a term which is discussed in more details later in subsection 2.1.4.3) relationships between independent and dependent variables, also they have the ability to detect all possible interactions between predictor variables and can be developed using multiple different training algorithms (Fausett, 2001).
- **Ease of use:** Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data (Haykin, 2001). Although the user does need to have some heuristic knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results.

Neural networks are applicable in virtually every situation in which a relationship between the predictor variables (independents, inputs) and predicted variables (dependents, outputs) exists, even when that relationship is very complex and not easy to articulate in the usual terms of "correlations" or "differences between groups" (Fausett, 2001). A few representative examples of problems to which neural network analysis has been applied successfully are:

- **Credit assignment:** A variety of pieces of information are usually known about an applicant for a loan. For instance, the applicant's age, education, occupation, and

many other facts may be available. After training a neural network on historical data, neural network analysis can identify the most relevant characteristics and use those to classify applicants as good or bad credit risks (Haykin, 2001).

- **Monitoring the condition of machinery:** Neural networks can be instrumental in cutting costs by bringing additional expertise to scheduling the preventive maintenance of machines (Mehrotra et al., 1996). A neural network can be trained to distinguish between the sounds a machine makes when it is running normally ("false alarms") versus when it is on the verge of a problem. After this training period, the expertise of the network can be used to warn a technician of an upcoming breakdown, before it occurs and causes costly unforeseen "downtime".

Due to the importance of ANNs as discussed above, the next subsections will concern about giving a general background of the biological model that inspired the development of ANNs and comparing it with the basic model of an ANN. Also the activation functions that are commonly used when building an ANN model are also discussed. Beside the common two types of ANNs and the training algorithms, the designing issues that are related to ANNs are also presented.

2.1.1 The Biological Model

The basic model of the artificial neuron is founded upon the functionality of the biological neuron. By definition, “Neurons are basic signaling units of the nervous system of a living being in which each neuron is a discrete cell whose several processes are from its cell body”.Each neuron is a specialized cell which can propagate an electrochemical signal. The neuron has a branching input structure (the dendrites), a cell body, and a

branching output structure (the axon). The axons of one cell connect to the dendrites of another via a synapse as shown in **Figure 2.1**. When a neuron is activated, it fires an electrochemical signal along the axon. This signal crosses the synapses to other neurons, which may in turn fire. A neuron fires only if the total signal received at the cell body from the dendrites exceeds a certain level (the firing threshold). The strength of the signal received by a neuron (and therefore its chances of firing) critically depends on the efficacy of the synapses. Each synapse actually contains a gap, with neurotransmitter chemicals poised to transmit a signal across the gap junction (Haykin, 2001).

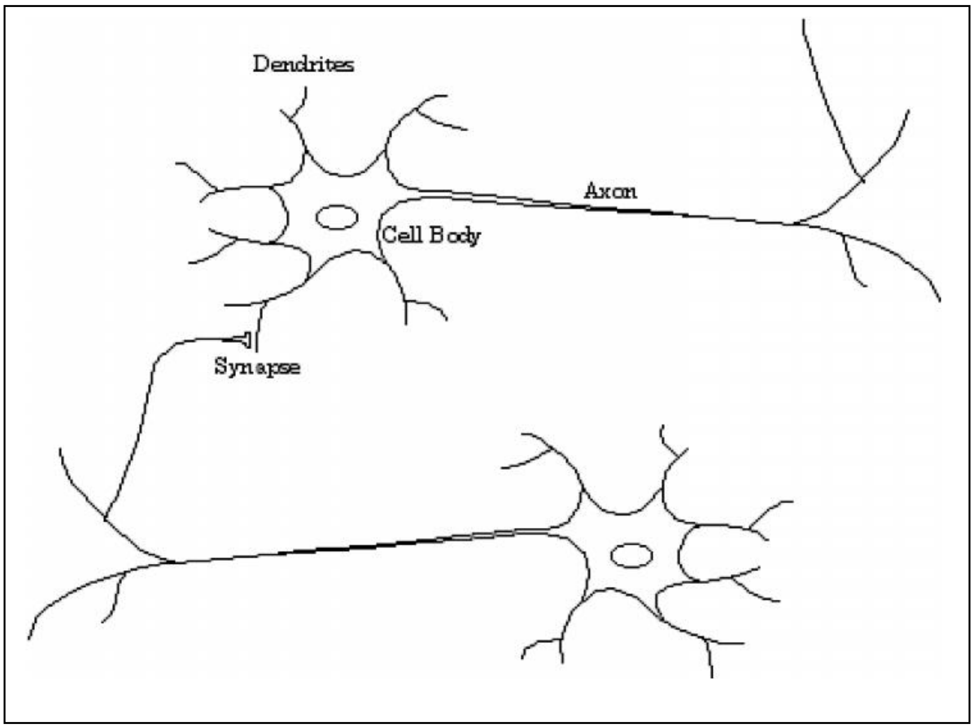


Figure 2.1: Biological Neuron (Haykin, 2001)

The brain which is principally composed of a very large number (10,000,000,000) of neurons, massively interconnected (with an average of several thousand interconnects per neuron, although this varies enormously) will analyze all

patterns of signals sent, and from that information it interprets the type of information received. Thus, from a very large number of neurons (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform extremely complex tasks (Haykin, 2001). Of course, there is a great deal of complexity in the brain which has not been discussed here, but it is interesting that ANNs can achieve some remarkable results using a model not much more complex than this.

2.1.2 The Basic Model

The basic model of ANN is an input layer consisting of nodes that simply accept the input values. The outputs of neurons in a layer are inputs to neurons in the next layer. The last layer is called the output layer. Layers between the input and output layers are known as hidden layers. **Figure 2.2** illustrates a diagram for this architecture.

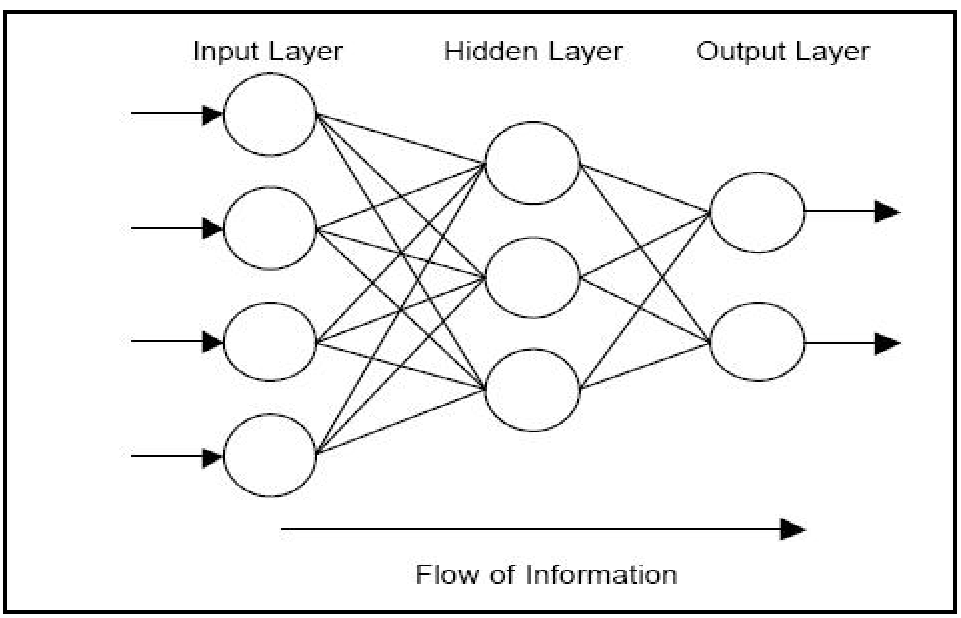


Figure 2.2: The basic model of an ANN

To capture the essence of biological neural systems, an artificial neuron is defined (Fausett, 2001) as follows:

- It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron.
- The weighted w_i sum of the inputs x_i is formed to compose the activation of the neuron. Gathering the weighs of the inputs according to equation (2.1).

$$Sum = \sum_{i=1}^{n} u_i w_i \qquad (2.1)$$

- The activation signal is passed through an activation function *(y)* (also known as a transfer function) to produce the output of the neuron as shown in equation (2.2).

$$y = f\left(\sum_{i=0}^{n} u_i w_i\right) \qquad (2.2)$$

Figure 2.3 illustrates a summary of the principle work of an artificial neuron model, and schematic illustrations of the basic features of a biological neuron and its artificial basic model is illustrated in **Figure 3.4**.

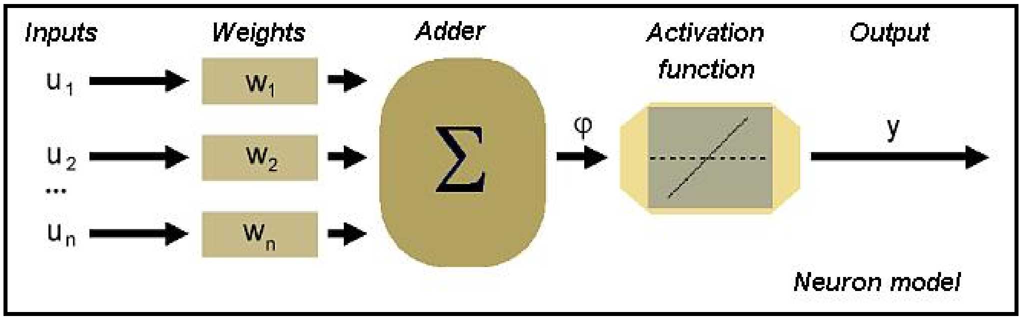


Figure 2.3: Summary of the principle work of Artificial Neuron Model(Fausett, 2001)

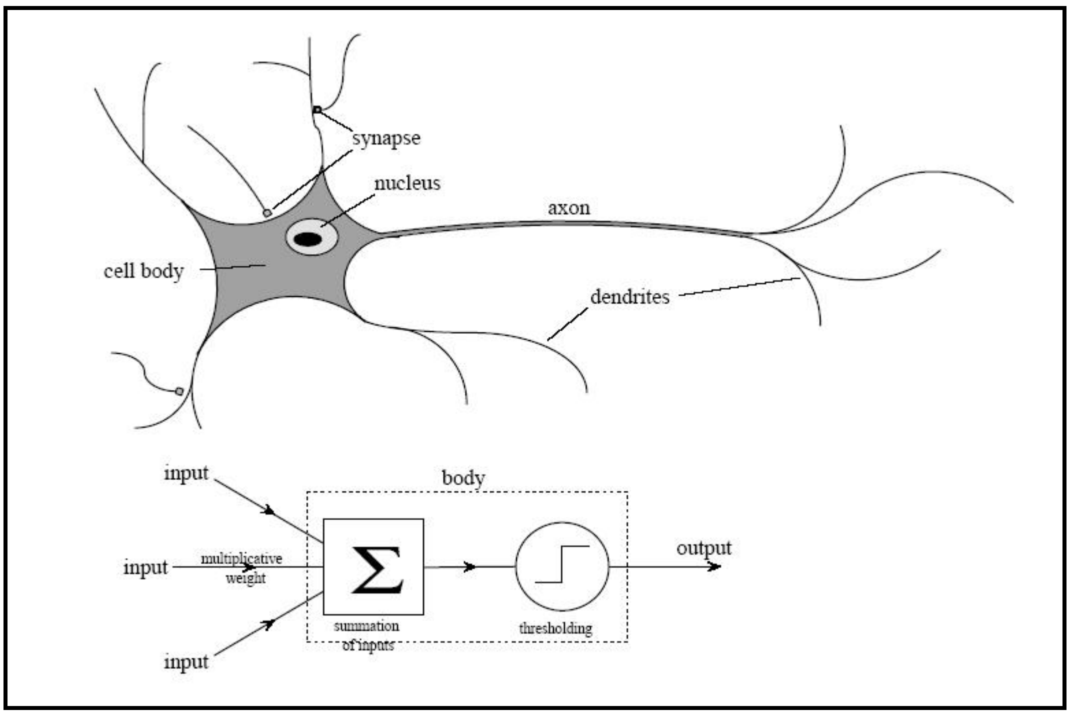


Figure 2.4 : Schematic Illustrations of the Basic Features of a Biological Neuron and its Artificial Basic Model(Fausett, 2001)

2.1.3 Types of Activation Function

The output of a particular artificial neuron is determined by the type of activation function, as known sometimes transfer function (Freeman and Skapura, 1991).An activation function is used to limit the amplitude of the output of a neuron and is commonly known as a squashing function (Haykin, 2001). Neural Networks supports a wide range of activation functions. The most common types of activation functions are:

1. **Linear activation function (LF)**: The outputs of a linear activation function, also known as a linear combiner, are determined as follows: $\boldsymbol{a} = \boldsymbol{n}$ where $\boldsymbol{n}$ is the sum of the weighted inputs. The graph of a LF function is represented as in **Figure 2.5**

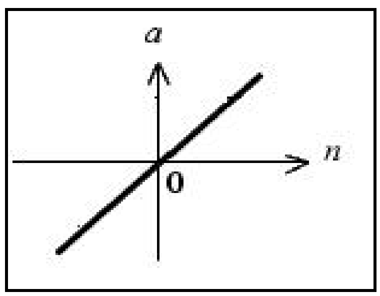


Figure 2.5: Graph of a Linear Activation Function

2. **Hyperbolic tangent sigmoid activation function (tanh)**: The hyperbolic tangent sigmoid activation function (tanh) has the following formula: $a = \frac{2}{1+e^{-2n}} - 1$

where , n is the sum of the weighted inputs. The resulted output from this function will be in a range between [-1,1] (Hagan et al., 1996). The graph of a tanh function is represented as in **Figure 2.6**

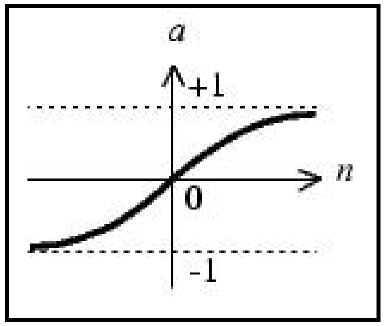


Figure 2.6: Graph of a Hyperbolic Tangent Sigmoid Activation Function

2.1.4 ANNs Architectures

ANNs architecture can be divided into two main categories based on their processing structure (known as topology): *feedback and feed-forward* neural networks (Haykin ,2001) (Mehrotra et al., 1996)**.**

In feedback neural networks, signals can travel in both directions. Thus the network has loops in its topology (Freeman and Skapura, 1991) as shown in **Figure 2.7(a).** Networks with this structure can be very complicated and powerful, as they can continuously change state until an equilibrium is reached – this balanced state is then kept up until changes in the input data occur (Mehrotra et al., 1996).

In feed-forward neural networks signals are allowed to travel one way only; from input to output as shown in **Figure 2.7 (b).** There are no feedback (loop) connections. Feed-forward networks tend to be straight forward models that associate inputs with outputs (Fausett, 2001).

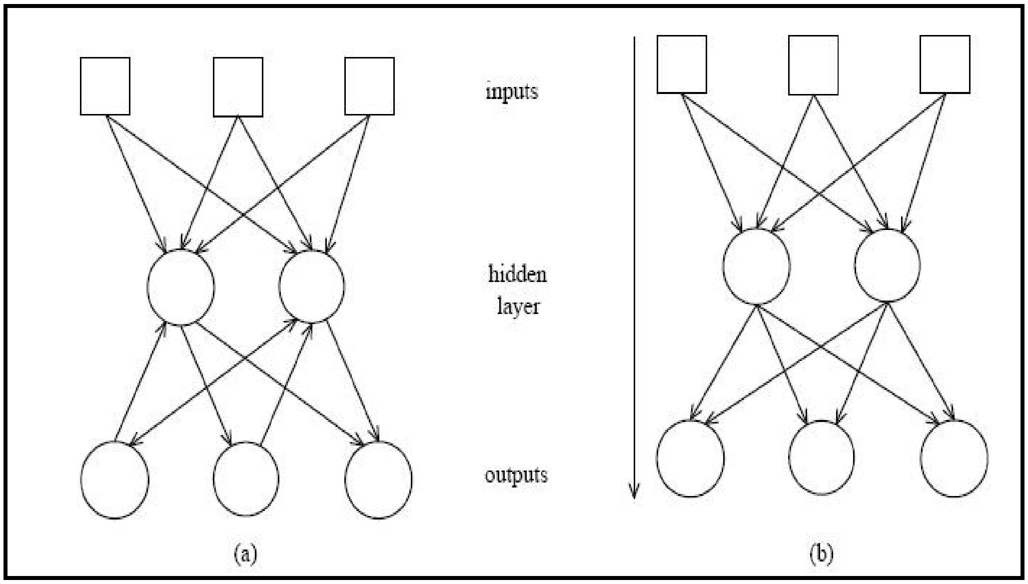


Figure 2.7: A (a) Feedback and a (b) Feed-forward Neural Network

These architectures can be further subdivided into two main classes based on the number of layers of processing nodes used in the model; single-layer perceptron and multi-layer perceptron (Fausett, 2001). In a single-layer perceptron there is a layer of input nodes and a layer of output nodes, which are the only processing nodes in the model. In a multi-layer perceptron (MLP) there are one or more hidden layers between the input and the output nodes.

2.1.4.1 The Single –Layer Perceptrons

In Single-Layer Perceptrons, or Perceptron, there is a layer of input nodes and a layer of output nodes, which are the only processing nodes in the model (Krose and Smagt, 1996). The general perceptron network architecture is shown in **Figure 2.8.**

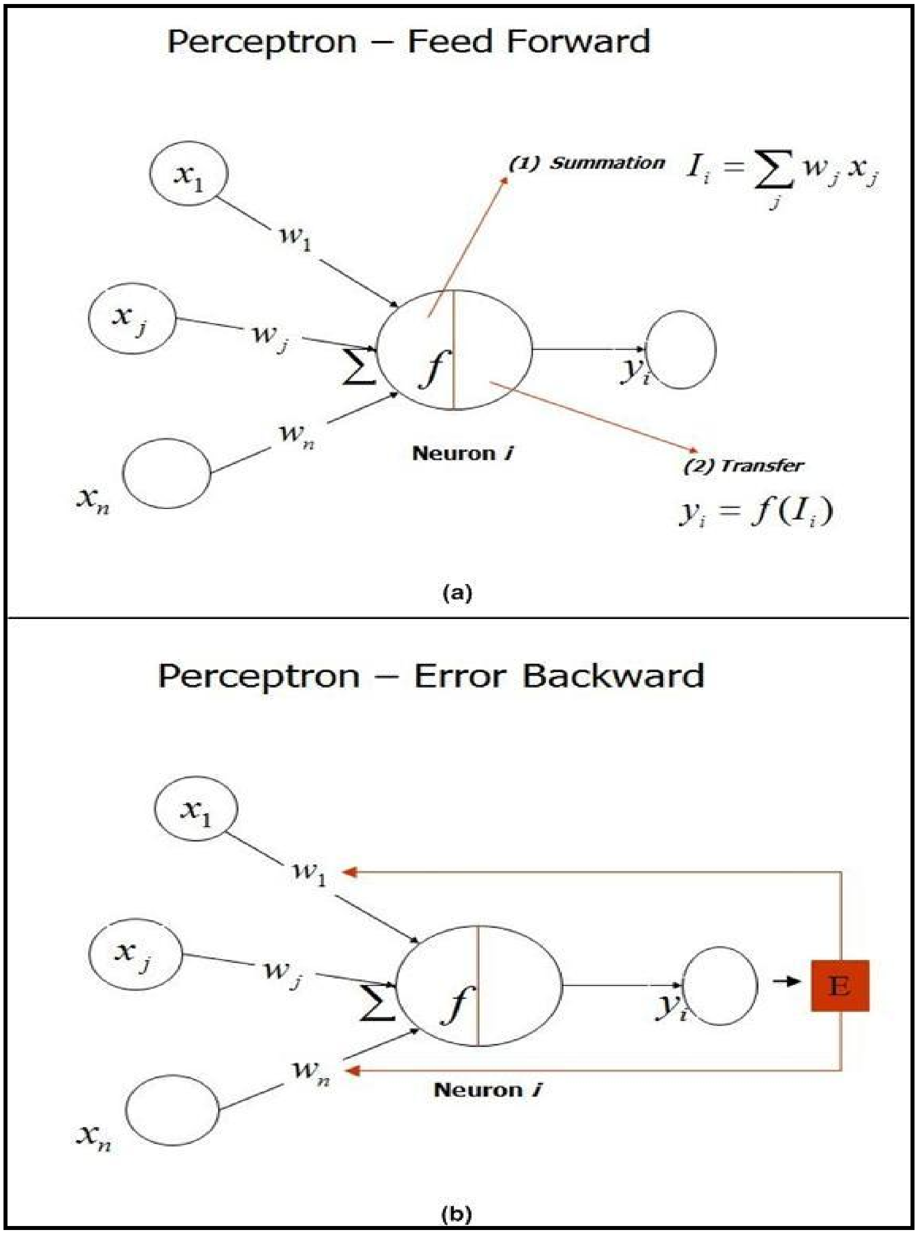


Figure 2.8: Feed-Forward Network with Single-Layer Perceptron (a) Feed Forward Stage, (b) Error Backward Stage

This network is one of the simplest ANNs whose weights and biases could be trained to produce a correct target vector presented with the corresponding input vector (Veropoulos, 2001). Perceptrons are especially suited for simple problems in pattern

classification (linearly separable problems). They are fast and reliable networks for the problems they can solve. This network is often trained with the perceptron learning rule. The perceptron learning rule is a learning rule for training single–layer hard–limit networks. It is guaranteed to result in perfectly functioning network in finite time (Hagan et al., 1996).

A perceptron neuron uses the hard-limit (threshold) transfer function, the hard-limit transfer function is one of the transfer functions that maps inputs greater than or equal to 0 to 1, and all other values to 0 (Freeman and Skapura ,1991). In a perceptron neuron (which uses the hard-limit transfer function (hardlim)) each external input is weighted with an appropriate weight $\boldsymbol{w_j}$, and the sum of the weighted inputs is sent to the hard-limit transfer function, which also has an input of 1 transmitted to it through the bias (Krose and Smagt, 1996).

The perceptron rule has two main stages; Feed-forward stage as in **Figure 2.8(a)**, and error backward stage as in **Figure 2.8(b)**, the details for each stage is as follow:

1. **Feed-forward stage**. Perceptron generates initially randomly weights, and then the weights ($\boldsymbol{w_j}$) and input values $\boldsymbol{x}_j$ are summed to compose activation function (v) as shown in equation (2.3) (Mehrotra et al., 1996).

$$I_i = w_0 + \sum_{j=1}^{m} w_j x_j \qquad (2.3)$$

Where, w_0 is called the bias that is a numerical value associated with the neuron which makes the neuron more powerful. It is convenient to think of the bias as the weight for an input x_0 whose value is always equal to one, so that the final formula become as equation (2.4).

$$I_i = \sum_{j=0}^{m} w_j x_j \qquad (2.4)$$

The perceptron neuron produces a 1 if the net input into the transfer function is equal to or greater than 0; otherwise it produces a 0.

2. **Error backward stage.** In this stage the output error is calculated as the following: $\varepsilon_i = t_i - y_i$, where t_i is the target and y_i is the output, if the error is equal to zero then no weight update is performed, else the weight must be updated according to equation (2.5) where Δw_j as shown in equation (2.6).

$$w_j^{(new)} = w_j^{(old)} + \Delta w_j \qquad (2.5)$$

Where $\Delta w_j = \varepsilon_i \eta x_j$ (2.6)

The term η is called the learning rate and is used to determine by how many the weights are changed at each step.

2.1.4.2 Multi-Layer Perceptron (MLP)

In MLP, or multilayer feed-forward networks, there are one or more hidden layers between the input and the output nodes (Machado, 1996) as shown in Figure 2.2. The nodes belonging to these layers are usually referred to as hidden nodes. A simple network has a feed-forward structure: signals flow from inputs, forwarded through any hidden units, eventually reaching the output units. Such a structure has stable behavior. In other words, the feed-forward network is a layered network in which each layer only receives inputs from previous layers (Mehrotra et al., 1996). The feed-forward structures have proved most useful in solving non-linearly separable problems (Haykin, 2001).

The input layer in an MLP contains units that simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer (Freeman and Skapura, 1991). When the network is executed (used), the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer (Fausett, 2001). The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

2.1.4.3 Linearly and Non-Linearly Separable Problems

A problem is said to be linearly separable; if a straight line can be drawn to separate the input vectors into two categories, if more than one line is needed to separate the input vectors then we said that this problem is non-linearly separable (Freeman and Skapura,

1991). As was said before, the single layer perceptrons are powerful to be used with linearly separable problems, while MLP are used with non-linearly separable problems.

A common example of a linearly separable problem is the AND Boolean function. **Table 2.1** is the truth table of the AND Boolean function, represented as in Figure 2.9, we can notice that the AND operation is a linearly separable problem because the separable line classified three points (0,0), (0,1) and (1,0) as one group by, while the point (1,1) was classified as another group.

Table 2.1: Truth Table of the AND Boolean Function

1	**X2**	**X1** *AND* **X2**
0	0	0
0	1	0
1	0	0
1	1	1

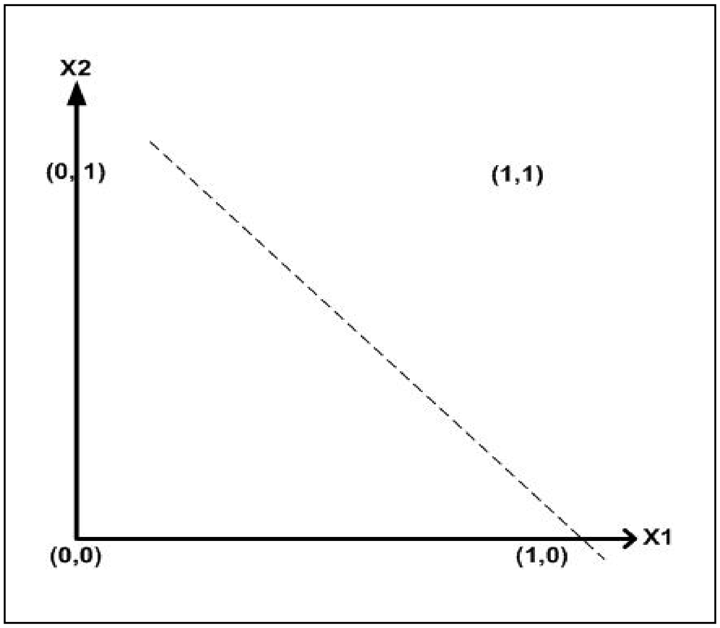


Figure 2.9: The AND Function is Linearly Separable

A common example of a non-linearly separable problem is the Boolean exclusive or (XOR) function, with the truth table shown in **Table 2.2** , the desired computation is ‘’yes (1)’’ when one or the other of input is on, and ‘’no (0)’’ when they are both off or both on. So as illustrated in **Figure 2.10**, there is no clear line that can separate the two types of points.

Table 2.2: Truth Table of the XOR Boolean Function

X1	**X2**	**X1** *XOR* **X2**
0	0	0
0	1	1
1	0	1
1	1	0

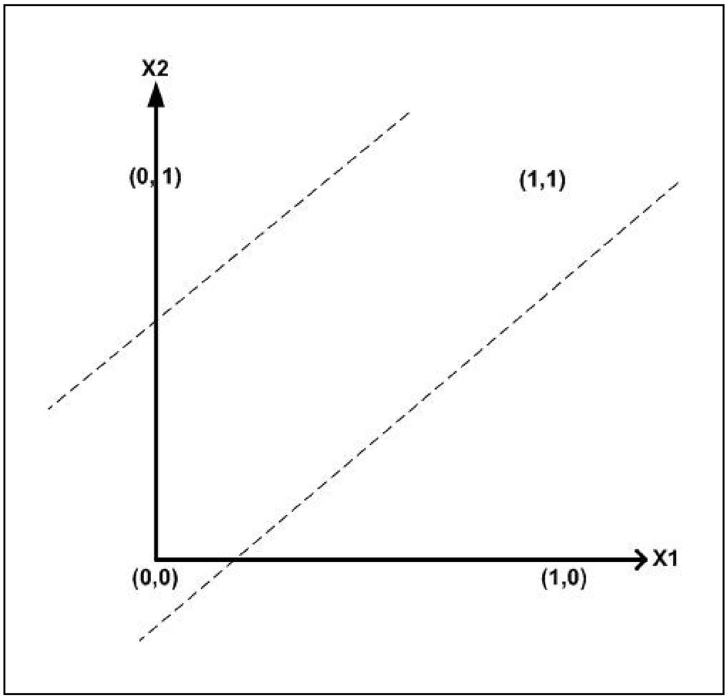


Figure 2.10: The XOR Function is Non-Linearly Separable

2.1.5 The Back Propagation Algorithm

MLP neural networks have been applied successfully to solve difficult and diverse problem by training them in a supervised manner with highly popular algorithm known as Back Propagation which uses the data to adjust the network's weights and biases in manner that minimizes the error in its predictions on the training set (Fausett, 2001). If the network is properly trained, it has then learned to model the (unknown) function that relates the input variables to the output variables, and can subsequently be used to make predictions where the output is not known.

A Back Propagation network learns by example. You give the algorithm examples of what you want the network to do and it changes the network's weights so that, when

training is finished, it will give you the required output for a particular input. In order to train the network you need to give it examples of what you want (the output you want (called the *Target*)) for a particular input as shown in **Figure 2.11**.

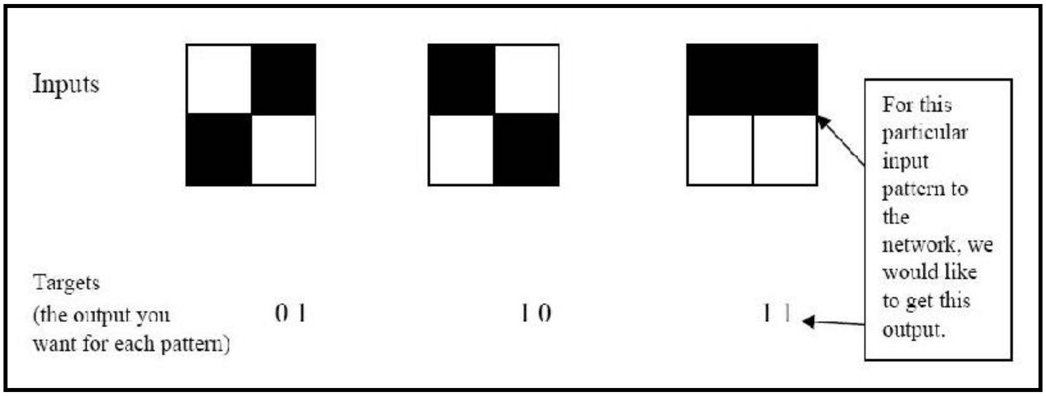


Figure 2.11: A Back Propagation Training Set(Krose and Smagt, 1996)

So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in **Figure 2.12** (a black pixel is represented by 1 and a white by 0). The input and its corresponding target are called a *Training Pair*. Once the network is trained, it will provide the desired output for any of the input patterns.

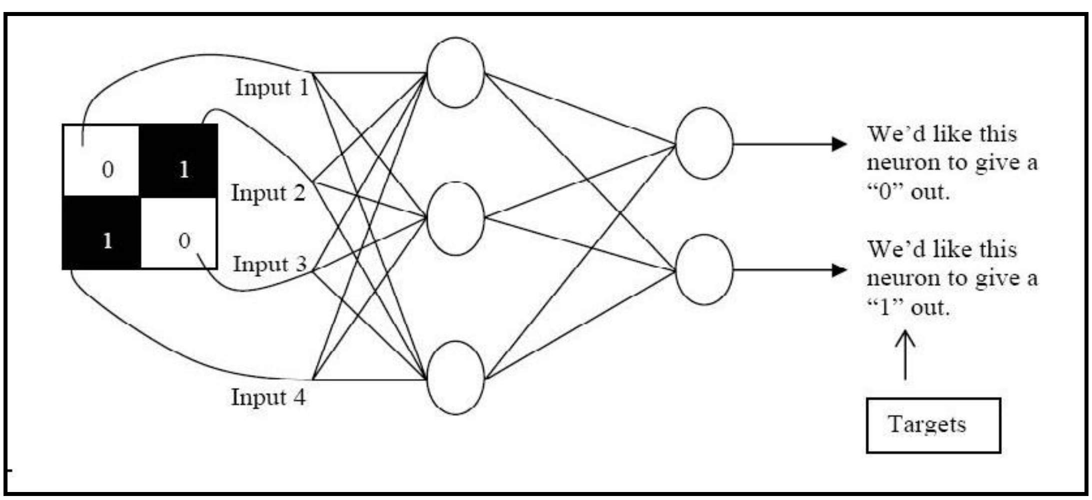


Figure 2.12: Applying a Training Pair to a Network. (Krose and Smagt, 1996)

The training works as follows (Krose and Smagt, 1996):

- The network is first initialised by setting up all its weights to be small random numbers.
- Next, the input pattern is applied and the output calculated (this is called the *forward pass*). The calculation gives an output which is completely different to what you want (the Target), since all the weights are random.
- We then calculate the *Error* of each neuron, which is essentially: *Target – Actual Output* (i.e. What you want – What you actually get).
- This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the *reverse pass*).
- The process is repeated again and again until the error is minimal.

2.1.5.1 The Error Surface

Since Back Propagation iteratively searches for a set of weights that minimizes the error function **(E)** over all training pairs. Therefore MLP networks have error surface, which is in general complex and believed to have many local and global minima. This occurs because the algorithm always changes the weights in such a way as to cause the error to fall. But the error might briefly have to rise, as shown in **Figure 2.13**. If this is the case, the algorithm will get stuck (because it can't go uphill) and the error will not decrease further (Fausett, 2001). This requires rerunning the algorithm with new random initial weights and other network parameter hoping to reach the global minima in the new run. As a consequence, Back Propagation is never assured to find the global minima.

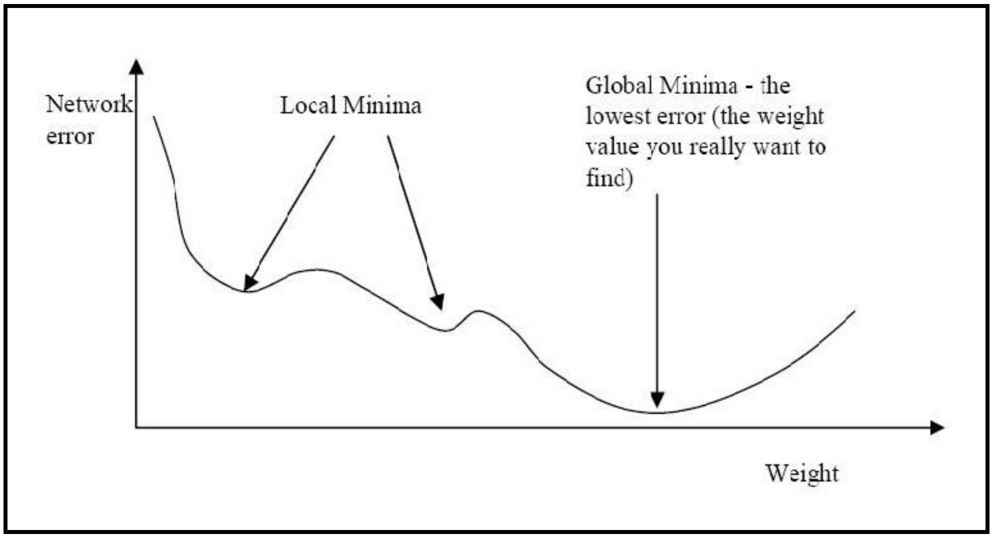


Figure 2.13: An Error Surface.

2.1.5.2 The learning Rate

An important consideration in the Back Propagation algorithm is the learning rate ***η*,** which determines by how much we change the weights w at each step. If ***η*** is too small, the algorithm will take a long time to converge (reaching an optimal set of weights that give the solution) (Hagan et al.,1996). The Influence of choosing a small learning rate on the error surface is illustrated in **Figure 2.14.**

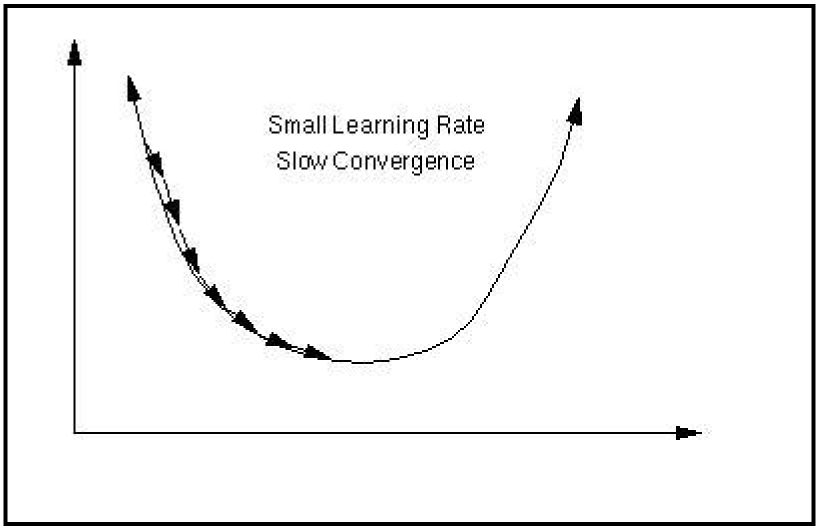


Figure 2.14: The Influence of Choosing a Small Learning Rate on the Error Surface

Conversely, if η is too large, we may end up bouncing around the error surface out of control, the algorithm diverges. The influence of choosing a large learning rate is illustrated in **Figure 2.15**. The correct setting for the learning rate is application-dependent, and is typically chosen by experiment; it may also be time-varying, getting smaller as the algorithm progresses (Fausett, 2001).

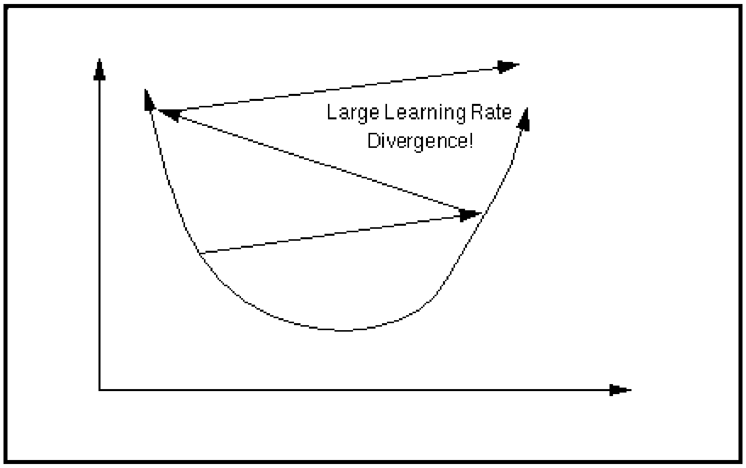


Figure 2.15: The Influence of Choosing a Large Learning Rate on the Error Surface

2.1.5.3 The Back Propagation Steps in Details

The architecture of Backpropagation as shown in **Figure 2.16** work accordingly to the following:

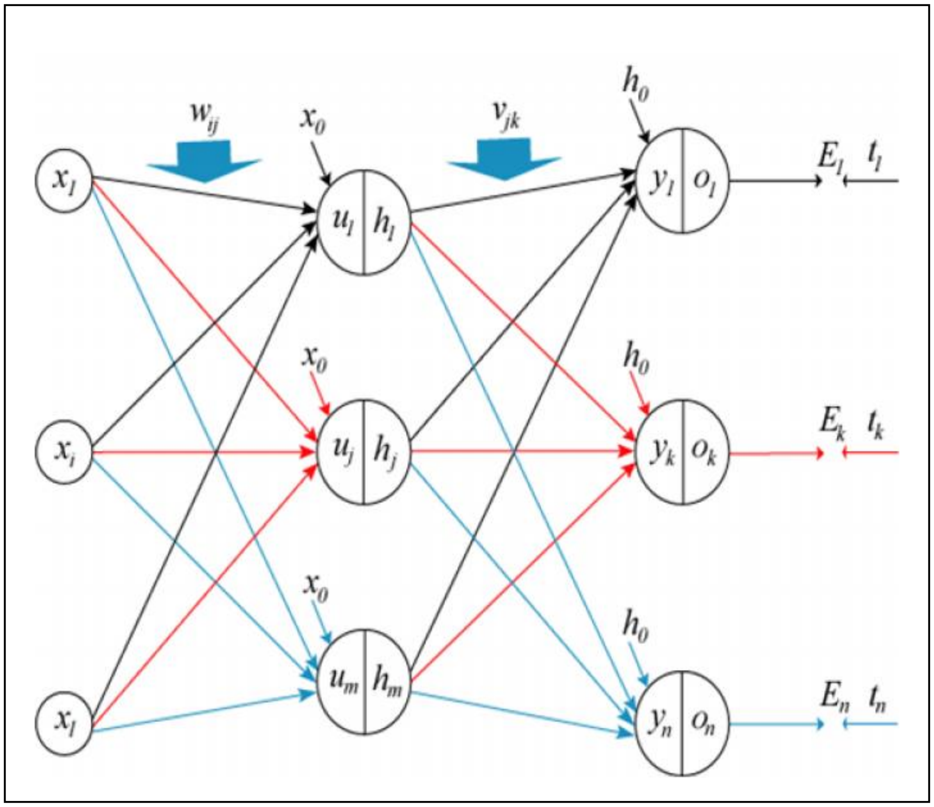


Figure 2.16: The architecture of Backpropagation

Step 1: Initialize the weights of the randomly with number between -1 and 1 according to equations (2.7) and (2.8).

$$w_{ij} = random\ ([-1,+1]) \qquad 0 \le i \le l, 1 \le j \le m \qquad (2.7)$$

$$v_{ij} = random\ ([-1,+1]) \qquad 0 \le j \le m, 1 \le k \le n \qquad (2.8)$$

Step 2: Present the pattern p to the network as shown in equation (2.9), where p is one pattern from the whole pattern P.

$$p = (x_1, x_2, x_3, x_i, \ldots. x_l) \in R^l \qquad p \in P \qquad (2.9)$$

Step 3: Compute the values of the hidden-layer nodes using the formula as shown in equation (2.10).

$$u_j = \sum_{i=0}^{l} w_{ij} \times x_i \qquad , h_j = f(u_j) \qquad ,1 \le j \le m \qquad (2.10)$$

Step 4: Calculate the values of the output nodes using the formula according to equation (2.11).

$$y_k = \sum_{j=0}^{m} v_{ik} \times h_j \qquad , o_k = f(y_k) \qquad ,1 \le k \le n \qquad (2.11)$$

Step 5: Compute the errors in the output layer using equation (2.12).

$$\delta o_k = (t_k - o_k) o_k (1 - o_k) \qquad ,1 \le k \le n \qquad (2.12)$$

Step 6: Compute the errors in the hidden layer using equation (2.13).

$$\delta h_j = h_j (1 - h_j) \sum_{k=1}^{n} \delta o_k \times v_{jk} \qquad ,1 \le j \le m \qquad (2.13)$$

Step 7: Adjust the weights between the output layer and the hidden according to equation(2.14).

$$v_{jk}^{(new)} = v_{jk}^{(old)} + \eta \times \delta o_k \times h_j \qquad ,0 \le j \le m, 1 \le k \le n \qquad (2.14)$$

Step 8: Adjust the weights between the hidden layer and the input layer according to equation (2.15).

$$w_{ij}^{(new)} = w_{ij}^{(old)} + \eta \times \delta h_j \times x_i \qquad ,0 \le i \le l \qquad ,1 \le j \le m \qquad (2.15)$$

Step 9: Repeat step 2 through 8 for each element of training set, until the Total Mean Squared Error (TMSE) ≤ E, E is an error threshold that is used to stop the training, error E defined as the difference between the desired t_k and current o_k output as shown in equation (2.16). (Hagan et al.,1996) :

$$TMSE = \frac{1}{N}\sum_{k=1}^{n}(t_k - o_k)^2 \qquad (2.16)$$

We notice from step 7 and 8 that these steps are the ones that change the weights of the network based on the general formula as shown in equation (2.17).

$$w_{ij}^{(new)} = w_{ij}^{(old)} + \Delta w_{ij} \qquad (2.17)$$

The term Δw_{ij}, is the amount of change in the current weight, this term in the back propagation algorithm is equal to; $\eta \times \delta o_i \times h_j$, where the term $\delta o_i \times h_j$ represent the partial derivative of the error with respect to the weight ($\frac{\partial E}{\partial w_{ij}}$) (Hagan et al., 1996).

The whole idea behind the partial derivative is; to gradually, but consistently, decrease the output error by adjusting the weights. The trick is to figure out how to adjust the weights. Intuitively, we know that if a change in a weight will increase (decrease) the error, then we want to decrease (increase) that weight. Mathematically, this means that we

look at the derivative of the error with respect to the weight: $\frac{\partial E}{\partial w_{ij}}$, once we find this derivative, we will update the weight via the formula as shown in equation (2.18).

$$\Delta w_{ij} = \eta \times \frac{\partial E}{\partial w_{ij}} \qquad (2.18)$$

2.1.6 Resilient Back Propagation Algorithm (Rprop)

As discussed in subsection 2.1.5.1 a Back Propagation algorithm will experience a local minimum that occurs because the algorithm always changes the weights based on the value of the partial derivative. So in order to solve this problem; the Resilient Back Propagation (Rprop) was devised (Riedmiller and Braun, 1993), where it eliminates the harmful influence of the magnitude of the partial derivative. As a consequence, only the sign of the derivative is considered to indicate the amount of the weight update. To achieve this, an update-value Δ_{ij} is introduced for each weight $\boldsymbol{w_{ij}}$, which solely determines the size of the weight-update (Riedmiller and Braun, 1993). This adaptive update-value evolves during the learning process**,** according to the equation (2.19) .

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \times \Delta_{ij}^{(t-1)} & , \ if \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} \times \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^{-} \times \Delta_{ij}^{(t-1)} & , \ if \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} \times \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)} & , \qquad else \end{cases} \qquad (2.19)$$

$$where \ 0 < \eta^{-} < 1 < \eta^{+}$$

The adaptation-rule works as follows: Every time the partial derivative of the corresponding weight $\boldsymbol{w_{ij}}$ changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_{ij} is decreased by the factor η^{-} (which was found by experiments that the best value for it is equal to **0.5** **).** If the derivative retains its sign, the update-value is slightly increased by the factor η^{+} (which was found by experiments that the best value for it is equal to **1.2**) in order to accelerate convergence in shallow regions.

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, if the derivative is negative, the update –value is added according to equations (2.20) and (2.21).

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & if \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)}, & if \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & , else \end{cases} \tag{2.20}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \tag{2.21}$$

However, there is one exception: If the partial derivative changes sign, i.e. the previous step was too large and the minimum was missed, the previous weight-update is reverted by formula as shown in equation (2.22).

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \quad if \frac{\partial E}{\partial w_{ij}}^{(t-1)} \times \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \qquad (2.22)$$

Due to that 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting $\frac{\partial E}{\partial wij}^{(t-1)} = 0$ in the Δ_{ij} adaptation rule above.

The update-values and the weights are changed every time the whole pattern set has been presented once to the network (learning by epoch).

Rprop Algorithm

The following pseudo-code fragment shows the kernel of the **RPROP** adaptation and learning process. The **minimum (maximum)** operator is supposed to deliver minimum (maximum) of two numbers; the **sign** operator returns +1, if the argument is positive, -1, if the argument is negative and 0 otherwise. While $\Delta_{min} = 10^{-6}$ and $\Delta_{max} = 50$.

For all weigths and biases {

if ($\frac{\partial E}{\partial w_{ij}}^{(t-1)} \times \frac{\partial E}{\partial w_{ij}}^{(t)} > 0$) **then** {

$$\Delta_{ij}^{(t)} = \textbf{minimum}\ (\Delta_{ij}^{(t-1)} \times \eta^{+}, \Delta_{max})$$

$$\Delta w_{ij}^{(t)} = -\textbf{sign}\ (\frac{\partial E}{\partial w_{ij}}^{(t)}) \times \Delta_{ij}^{(t)}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

}

else **if** ($\frac{\partial E}{\partial w_{ij}}^{(t-1)} \times \frac{\partial E}{\partial w_{ij}}^{(t)} < 0$) **then** {

$$\Delta_{ij}^{(t)} = \textbf{maximum}\ (\Delta_{ij}^{(t-1)} \times \eta^{-}, \Delta_{min})$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \Delta w_{ij}^{(t-1)}$$

$$\frac{\partial E}{\partial w_{ij}}^{(t)} = 0$$

}

else **if** ($\frac{\partial E}{\partial w_{ij}}^{(t-1)} \times \frac{\partial E}{\partial w_{ij}}^{(t)} = 0$) **then** {

$$\Delta w_{ij}^{(t)} = -\textbf{sign}\ (\frac{\partial E}{\partial w_{ij}}^{(t)}) \times \Delta_{ij}^{(t)}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

}

}

2.1.7. Designing Issues of Artificial Neural Networks

There are several issues to be considered when designing a neural network application, aiming to achieve a good generalization performance – i.e. to have the outputs of the network approximate well target values given inputs that are not in the training set. These issues will be discussed in the following subsections.

2.1.7.1. Selection of an ANN Topology

When it comes to decide what ANN topology to be used and which training algorithm to apply to this structure, there is no easy answer. Selection of a suitable combination of these two design components is often done experimentally and it is problem dependent (Mehrotra et al., 1996).

Other issues closely related to the network topology include the number of hidden layers and hidden units to be used. In a wide variety of applications a linear model can be sufficient for developing a good classification system. In that case no hidden layers will be needed (Fausett, 2001). However, in applications where a nonlinear model is needed, a MLP with one hidden layer with arbitrary large number of units typically suffices for a good approximation. The optimum number of hidden units depends on the number of input features, the size of the training set, the amount of noise presented in the data, the architecture and the choice of functions and parameters in the algorithm (Veropoulos, 2001). Although a lot of attempts have been made in order to determine a general rules that give the most appropriate number of hidden units to be used for training the network (Mehrotra et al., 1996), none of them can be considered reliable enough or true for any situation.

Another issue is to determine how many training samples are required for successful learning, and how large a neural network is required for a specific task, these are solved in practice by trial and error (Mehrotra et al., 1996). These issues are complex because there is considerable dependence on the specific problem being attacked using a neural network (Veropoulos, 2001).

2.1.7.2. Pre-Processing and Post-Processing

The input data should be in such a form that a neural network can process it, this means numeric values. Furthermore, the numeric input values may be pre-processed in a way that the network's learning task is easier. In many practical applications, the choice of the pre-processing method will be one of the most significant factors in determining the performance of the ANN (Hagan et al., 1996). Pre-processing is always performed before the training process begins. Sometimes the output data are also processed. This is called post-processing which is converting back the output into the same units that were used in the original dataset.

One commonly used pre-processing method is data standardization; standardizing a vector most often means subtracting a measure of location and dividing by a measure of scale. For example, you might subtract the mean and divide by the standard deviation, thereby obtaining a "standard normal" random variable with mean 0 and standard deviation 1.

The average (mean) is a measure of the middle value of the data set. If n numbers are given, each number denoted by $\boldsymbol{a_i}$, where i = 1,..,n, the mean is the [sum] of the $\boldsymbol{a_i}$'s divided by n or by the formula as shown in equation (2.23).

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{N} a_i \qquad (2.23)$$

While the standard deviation shows how much variation or "dispersion" there is from the "average" (mean). A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data are spread out

over a large range of values (Ghahramani, (2000). The standard deviation $\boldsymbol{S_N}$ is defined as shown in equation (2.24).

$$S_N = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(x - \bar{x})^2} \qquad (2.24)$$

In order to obtain a dataset with zero mean and unit standard deviation, we use the following:

$\boldsymbol{X_i}$ = Value of the raw input variable X for the $\boldsymbol{i}^{\boldsymbol{th}}$ training case.

$\boldsymbol{S_i}$ = Standardized value corresponding to $\boldsymbol{X_i}$.

$\boldsymbol{N}$ = Number of training cases.

Standardize X to mean 0 and standard deviation 1 by equation (2.25).

$$\boldsymbol{S_i} = \textbf{(Xi – mean(X)) / Standard Deviation(X).} \qquad (2.25)$$

Note that statistics such as the mean and standard deviation are computed from the training data, not from the validation or test data. The validation and test data must be standardized using the statistics computed from the training data.

2.1.7.3. Generalization Performance of an ANN

One of the major advantages of neural networks is their ability to generalize; this means that a trained network could classify data that it has never seen before from the same class as the learning data. In real world applications developers normally have only a small part of all possible patterns for the generation of a neural network (Hagan et al., 1996). To reach the best generalization, the dataset should be split into three parts:

- **The training set** that is used to train a neural network. The error of this dataset is minimized during training.
- **The validation set** that is used to determine the performance of a neural network on patterns that are not trained during learning.
- **A test set** for finally checking the over all performance of a neural network.

The learning should be stopped in the minimum of the validation set error. At this point the network generalizes best. When learning is not stopped, overfitting occurs and the performance of the network on the whole data decreases, despite the fact that the error on the training data still gets smaller. After finishing the learning phase, the network should be finally checked with the test set (Veropoulos, 2001). An example of an overfitted network is illustrated in **Figure 2.17**

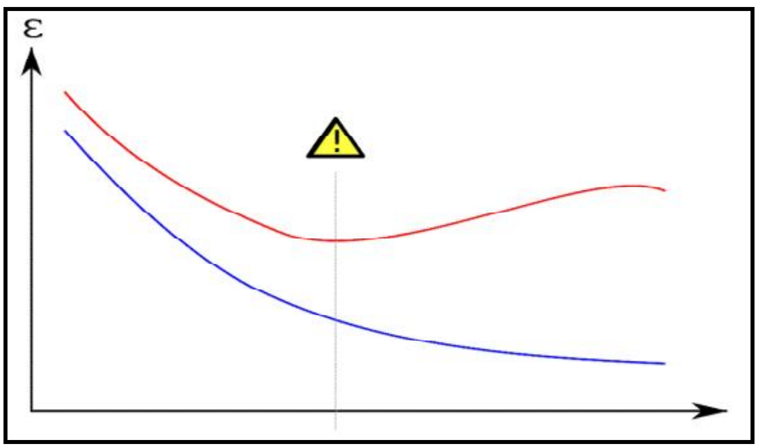


Figure 2.17: Overfitting in Neural Networks, Training error is shown down and validation error is shown up

This technique is called early stopping and it is used for improving the generalization of a neural network during training and preventing it from overfitting (Haykin 2001). Where the basic concept is that the performance of the ANN against the training set will

continue to improve with more training, and this is also true for its performance against the validation set (Machado1996). However, if the ANN starts overfitting the training set, its performance then against the validation set will start to decrease. Therefore, the validation error will increase; this is an indicative of a poor generalization and can be used for stopping the classifier from further training (Haykin 2001). The weights giving best performance against the validation set can then be accepted as the set of weights that gives the ANN best generalization ability. Since the error surface is not always smooth, it is not certain at this point of the training if the validation error is the global minimum (Hagan et al., 1996). So, it is common practice to continue training for several iterations (based on the judgment of the designer) in case the ANN performance against the validation set improves again. At the same time the configuration and set of parameters of the ANN that gave best results until this point are always stored (Veropoulos, 2001).

2.2. Using Artificial Neural Networks in Medical Problems

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns that are too complex to be noticed by either humans or other computer techniques, therefore, due to their capabilities they have been used widely in the medical field. Section 2.2.1 discusses some of the applications of neural networks in the medical domain in general, where section 2.2.2 presents some of the applications of neural networks in gynaecology domain.

2.2.1. Applications of Neural Networks in the Medical Field

Previous studies sought ways of capitalizing the use of neural networks in the diagnosis of breast cancer to improve the results of breast cancer screenings by evaluating the performance ANN to predict an outcome (cancer/not cancer). Paulin and Santhakumaran (2011) presented a study on the classification of breast cancer using feed forward ANNs. the performance of the network is evaluated using Wisconsin Breast Cancer Data set **(WBCD)** where an accuracy of 99.28% using the Levenberg-Marquardt **(LM)** training algorithm was obtained. Also (Kıyan and Yıldırım, 2003) evaluated the performance of the Radial Basis Network **(RBF)**, General Regression Neural Network **(GRNN)**, MLP and Probabilistic Neural Network **(PNN)** on the WBCD and they concluded that the most suitable neural network model for classifying WBCD is GRNN with accuracy 98.8% while the classification performances for the other type of neural network were 96.18% for RBF, 97.0% for PNN, and 95.74% for MLPs.

Bascil and Temurtas, (2009) presented a study for hepatitis diagnosis using a multilayer neural network as a machine learning technique and LM algorithm as training algorithm, the architecture was on the data set that was taken from University of California Irvine **(UCI)** machine learning database, the study obtained a classification accuracy of 91.87% via ten-fold cross validation.

On the other hand, there some studies showed the ability of neural networks in predicting diseases complication. Vahdani et al. (2009) considered in their study on predicting the presence or absence of cirrhosis in patients with chronic hepatitis B; they

concluded that in patients with chronic hepatitis B, if the ANN value of certain laboratory manifestations is negative, there will be a 95% chance of having a negative liver biopsy.

In the field of diabetes diagnosis; (Kayaer and Yildirim, 2003) applied two different neural network structures, which are (MLP) and (GRNN) to the Pima Indians Diabetes **(PID)** medical data obtained from the UCI machine learning repository. They achieved 80.21% classification accuracy using a GRNN and they have also reported 77.08% classification accuracy using multilayer neural network with LM training algorithm. Furthermore (Temurtas et al., 2009) in their study realized a PID diagnosis system; for this purpose, a multilayer neural network structure which was trained by LM training algorithm was used, the study achieved 79.62 % accuracy using ten-fold cross validation approach and it achieved 82.37% classification accuracy using the conventional validation approach (one training set and one test set).

In medicine there are a several diseases that share the same symptoms which increase the ability of wrong diagnosis such as heart diseases so that machine learning can help in making correct diagnoses. Yan et al. (2006) presented a multilayer perceptron based decision support system to support the diagnosis of heart diseases; the system was applied to five different heart diseases. The results showed that the proposed system could achieve very high diagnosis accuracy 90% proving its usefulness in support of clinical diagnosis decision of heart diseases

AlTimemy and Al Naima (2010) in their study presented the prediction of kidney dysfunction using different neural network approaches, PNN, SOM and MLP neural network trained with back propagation algorithm; these networks are used as a classifiers to

predict whether kidney is normal or it will have a dysfunction applied on a data set collected from one of the private clinical laboratories in Baghdad, In this study the PNN gave more accurate prediction of kidney dysfunction with accuracy 99% while the accuracy of SOM was 98% and the accuracy for MLP was 95%.

2.2.2. Applications of Neural Networks in the gynaecology Domain

The gynecology domain have many diseases that have varies and complex symptoms, hence There are not many research in this specific domain, and there is not specific research lies any disease of gynecological diseases for diagnoses except cancer, all research presented only one disease of gynecology such as ovarian cancer or breast cancer .

Mangalampalli et al. (2006) in their study presented system for treatment gynecological diseases, to develop a decision-support system which can help a gynecologist analyze and prescribe medicines for such diseases with fewer errors. This system prescribes a medicine for a gynecological disease based on various symptoms of the disease. That implemented based on the multi-layered feed forward neural network structure. This system is almost the only one of its kind which deals with more than disease in one system. It is dealing cervix symptoms, uterus symptoms and others, this system deals with gynecological diseases in terms of description of treatment does not classification of the diagnosis.

Also, (Wilding et al., 1994) proposed two systems to diagnose two types of cancer are breast cancer and ovarian cancer using a group of laboratory tests and the age of the patient. These methods provide an objective approach to backpropagation neural networks,

the breast cancer study involved 104 patients (45 malignant and 59 benign subjects) and achieved accuracy 72.9% , While the ovarian cancer study involved 98 individuals (35 malignant, 36 benign and 27 control subjects) and achieved accuracy 85.5%. (Menéndez et al ,2010) in this paper describes a neural network based approach to breast cancer diagnosis; the model developed is able to determine which women are more likely to suffer from a particular kind of tumor before they undergo a mammography.

While (Szpurek et al .2005) develop in their study artificial neural network model, this study include the 686 women, 431 (62.8%) had a benign and 255 (37.2%) had a malignant ovarian tumor, and achieved accuracy 97.7%.

Chapter Three

3. Data Preparation and System Architecture

ANN performs quit well with complex tasks such as classification problems using abstract data. ANNs have been applied to many areas of the medical field for analysis of various diseases and conditions. Application of ANN to the diagnosis for precise types of gynaecological diseases is a fairly new area, and not much analysis has been performed with ANNs. In this chapter section 3.1 illustrates how data was equipped for system's training and testing processes. Section 3.2 concerns about the process of choosing the best system architecture for the CDSSs.

3.1. Data Preparation

When a patient reports to a physician, a large number of possibly relevant inputs must be considered during the diagnosis. A physician reaches an accurate diagnosis or treatment decision based upon observations, the patient's answers to questions and physical examinations.

In this thesis, the heart disease dataset used for testing and training the system consisted of a total of 550 cases (in which each disease has a total of 50 cases) gathered from the gynecology clinics at JUH, located in Amman, Jordan. Moreover, 54 variables essential to the diagnosis of the gynecological diseases of interest were interpreted from each patient's medical record. These variables can be divided into two categories:

i. Basic information of a patient (including age, marital status and pregnant).

ii. Symptoms (51 factors in total), can be divided into six categories in which are menstrual problems, vaginal problems, sexual problems, urinary and abdomen problems, pain problems and finally another problems.

In **Figure 3.1** the datasheet that was used in gathering the patient's variables is illustrated; whereas we can notice from the figure that most of the attributes were assigned to have a yes or no value to indicate the presence or absence of an attribute except for the following attributes:

- Appetite attribute (that belongs to the patient's symptoms category in another problems), one of three values will be assigned; an increase appetite or a normal appetite or a decrease appetite.
- Regular menstruation attribute (that belongs to the patient's symptoms category in menstrual problems), one of three values will be assigned; an absent menstruation or a regular menstruation or an irregular menstruation.
- Amount of bleed attribute (that belongs to the patient's symptoms category in menstrual problems), one of four values will be assigned; an absent or a spotting or a normal or heavy.
- One value was assigned for the marital status; either single or married.
- The age was assigned the actual values the patient has for this attributes.

Patient's Datasheet

➢ **Disease Type:**

1.Polyps 2.Infection 3.Migrans 4.Prolapse 5.Fibroids 6.Cancer
7.Endometrial hyperplasia 8.Amenorrhea 9. Abortion 10.Dysmenorrhea 11. Infertility

➢ **Basic information:**

– Age: _______ - Marital Status: *Single / Married.* - Pregnant: Yes/*No.*

➢ **Symptoms:**

❖**Menstrual Problems** :

– Menopause: *Yes/No.* - Regular menstruation: *Absent/Irregular/Regular.*

– Cramps menstruation: *Yes/No.* - Amount of bleed: *Absent/Spotting/Normal/Heavy.*

- Intermenstrual bleeding: *Yes/No.* - Pain before and/or after menstruation: *Yes/No.*

- Extended menstruation : *Yes/No.* - Postmenopausal bleeding: *Yes/No.* -Painful menstruation : *Yes/No.*

❖**Vaginal Problems:**

– Something protruding out of the vaginal: *Yes/No.* - Abnormal vaginal bleeding: *Yes/No.* -Vaginal mass: *Yes/No.*

– Yellow non-smelling vaginal discharge: *Yes/No* - Abnormal vaginal discharge: *Yes/No.*

– Yellow smelling vaginal discharge: *Yes/No.* - Vaginal discharge containing blood: *Yes/No.*

– Pressure symptoms with long standing: *Yes/No.* – Reduction in symptoms when lying down: *Yes/No.*

❖**Sexual Problems** :

– Bleeding after sexual intercourse: *Yes/No.* - Sexual dysfunction: *Yes/No.*

– Bleeding during sexual intercourse: *Yes/No.* – Pain during sexual intercourse : *Yes/No.*

❖**Urinary and Abdomen Problems :**

– Urinary frequency : *Yes/No.* -Urinary incontinence: *Yes/No* - Difficulty emptying the bowel : *Yes/No*

– Difficulty stopping urination : *Yes/No* - Difficulty starting urination : *Yes/No* - Increased urge to urinate : *Yes/No*

- Dragging sensation in lower abdomen : *Yes/No* – Difficult or painful urination : *Yes/No* - Feeling of rectal fullness : *Yes/No*

– A feeling of fullness in the lower abdomen : *Yes/No* - Painful bowel with cycles : *Yes/No* -Intestinal distress : *Yes/No*

❖ **Pain Problem:**

- Bladder pain: *Yes/No* – Pelvic pain: *Yes/No* – Low back pain: *Yes/No* -Loin pain: *Yes/No*

- Valur pain : *Yes/No* - Painful orgasms: *Yes/ No* - Lower abdominal pain : *Yes/No*

❖ **Another Problems**

- Redundant breast: *Yes/No* - Abnormal hair growth mainly at face: *Yes/No* - Fatigue: *Yes/No* - Dry skin: *Yes/No*

– Male factor: *Yes/No* - Endocrine dysfunction: *Yes/No* -Fever: *Yes/No* - Trauma: *Yes/No*

– Appetite: *Increase/ Decrease /Normal* - Thyroid deficiency : *Yes/No*

Figure 3.1: Patient's Datasheet

3.2. System Architecture

The following subsections explain how the input variables that were used in the training and testing process were encoded for building the gynecological disease CDSSs, and how the number of hidden layers and hidden neurons in each layer were chosen.

3.2.1. Input Variables Encoding Schemes

All neural networks take numeric input and produce numeric output. Therefore the 54 variables were encoded into numeric values, as well as the gynecological disease's classes using the following encoding schemes:

i. Numerical variables such as age were reserved the same with no changes since they are already in a numerical format.

ii. Variables with two independent attributes, such as the marital status and pregnant, symptoms are encoded with binary values (0, 1). For instance, 0 represents single and 1 married; 1 is adopted for the mentioned categories for the presence of the attribute in that category, and 0 for the absence of it.

iii. Variables with three independent attributes, such as Regular menstruation and appetite are encoded using the ternary values (0, 0.5, 1); with in regular menstruation attribute 0 representing that the patient is absent menstruation, 0.5 representing an irregular menstruation patient and 1 a regular menstruation patient. While in appetite attribute 0 representing that the patient is decrease appetite patient, 0.5 representing a normal appetite patient and 1 an increase appetite patient.

iv. Variables with four independent attributes, such as amount of bleed are encoded using the quaternary values (0, 0.25, 0.5, 1); with 0 representing that the patient is absent menstruation, 0.25 representing a spotting amount menstruation, 0.5 representing a normal amount menstruation patient and 1 a heavy amount menstruation patient.

After encoding all variables, the training dataset was standardized to have a zero mean and a unit standard deviation as explained in subsection 2.1.7.2, and based on the information that was collected form the training dataset during the standardizing; the validation and test datasets were standardized also to have a zero mean and a unit standard deviation.

3.2.2. Number of Hidden Layers and Hidden neuron

Optimizing the number of hidden layer and neurons in each hidden layer for a feed forward neural network remains one of the unsolved tasks in this research area. Thus several researchers have proposed some general rules for determining an optimal number of hidden neurons for any application. One of the rules is: one hidden layer is enough, and to find the number of neurons for the first hidden layer; add the number of neurons in the input layer to the number of neurons in the output layer and divide the result by two. Another known rule is: if more than one hidden layer must be used to achieve a less generalization error; then the number of neurons in each hidden layer except the first one will be the number of neurons in the previous layer divided by two (Xu and Chen, 2008). Moreover, Kavzoglu as cited in (Yan et al., 2006) has shown that in most situations, there is no way to determine the best number of hidden layers without training several networks and estimating the generalization error of each, because the best number of hidden layers

and units depends mainly in a complex way on: the number of input and output units, the number of training cases and the complexity of the classification problem to be learned (Xu and Chen, 2008).

Based on this, we used an iterative process to determine the best number of hidden layers and neurons in each hidden layer. In the iterative process a ten- fold cross validation technique (more details will be provided in chapter 4) is used to choose the best neural model that will provide the highest classification accuracy, the whole process works as follow:

Step1. Start testing the network architecture with only one hidden layer; the formula as shown in equation (3.1) applied to find the number of neurons in the first hidden layer.

$$n_f = (n_i + n_o)/2 \qquad (3.1)$$

Where n_f is the number of neurons in the first hidden layer, n_i is the number of neurons in the input layer and n_o is the number of the neurons in the output layer. If n_f was not a fixed number then apply the ceil and the floor operations, so you will get two values, and for more precise results take another number which is floor(n_f)-1, so you will have three numbers of neurons in the first layer to start with.

Step2. Add another layer; the number of neurons in this layer will be half the number of neurons in the previous layer.

Step3. Repeat step two until the number of the neurons in the layer is one.

Chapter Four

4. Experimental Results and Evaluation

In this chapter, Section 4.1 introduces the evaluation method that was used to assess the generalization of the ANN models. Section 4.2 presents multiple ANN architectures that were built in order to choose the best ANN model for building the CDSS for the gynecological diseases diagnosis. Finally Section 4.3 introduces the detailed results for the best ANN model we obtained in Section 4.2.

4.1. Cross Validation

Cross validation is a technique used for estimating the performance of a predictive model, also it can be referred as estimate of accuracy, sometimes referred as rotation estimation, and it is determined by the overall number of correct classifications divided by the total number of instances in the dataset (Yan et al., 2006). One round of cross-validation involves partitioning a sample of data into subsets, performing the analysis on one subset (called the training set), and testing the analysis on the other subset (called the testing set). As we discussed before in subsection 2.1.7.3, ANN go through overfitting, therefore a third subset called the validation set is used to avoid overfitting. Moreover to reduce variability and to avoid bad splits that may produce overfitting, multiple rounds of cross-validation are performed using different partitions, and the tested results are averaged over the rounds.

In this thesis a ten-fold cross validation was used to access the generalization of the network model that was built. A percentage of 80%, 10% and 10% data split of the complete data samples was applied to represent the train, validation and test subsets respectively(Yan et al., 2006). This provides for each fold a total of 440, 55 and 55 sample

cases for the train, validation and test subsets respectively. Each disease in the training set will be represented by 50 samples of the total samples of the training set. Also each disease in the validation set will be represented with 5 samples and 5 samples for the test set.

Classification Accuracy

In order to find the classification accuracy for a model, we must first calculate the classification accuracy for each individual fold, and after that averaging the result over the ten folds as follows:

- Compute the classification accuracy for each fold; the classification accuracy $\boldsymbol{A_{fi}}$ of an individual fold $\boldsymbol{f_i}$ depends on the number of samples correctly classified and is evaluated by the formula in equation (4.1).

$$A_{f_i} = (t / n) \times 100\% \qquad (4.1)$$

  Where t is the number of sample cases correctly classified, and n is the total number of sample cases.

- Find the average classification accuracy for the ten folds as shown in equation (4.2).

$$\text{Average Classification Accuracy} = \frac{1}{10} \sum_{i=1}^{10} A_{f_i} \qquad (4.2)$$

4.2. Models' Building

In order to investigate the best system architecture for building the CDSSs for the gynecological diseases diagnosis; multiple system architectures were built using the iterative process that was explained in subsection 3.2.2. The different architectures were compared based on the classification accuracy each architecture has obtained. The model with the highest classification accuracy (less generalization error) was chosen as the final model for building the CDSSs.

The different ANN architectures have been built and tested using MATLAB® version 2010a, which was run under Windows XP Professional SP3 OS with a Celeron Intel processor of 2.52 GHz, and 1.50 GB of RAM.

In this thesis for all models:

- A Feedforward Backpropagation neural network is used for building all models.
- The number of neurons in the input layer is 54 (representing number of features for each sample).
- The number of neurons in the output layer is 1 (representing one class of the diseases the neural will generate).
- The training algorithm that was used for training the models is the Rprop.
- LF and TANH are the activation function that were used in the output layer for each architecture, and the results obtained using both functions were compared.
- The activation function that is used for all the hidden neurons in all the hidden layers is TANH.

- The following values were used 0.07, 0.001 and 48 for the delta initial, the performance goal error and the number of validation checks to avoid the overfitting of the network respectively.

4.2.1. Models' Building using One Hidden Layer

Since the input layer contains 54 neurons and the output layer contains 1 neuron, the number of neurons in the first hidden layer will be 27.5 based on the equation in step 1 from the iterative process. Because the result is not a fix number, applying the ceil operation will produce 28, applying the floor operation will produce 27, therefore the network can be trained with two different numbers of neurons, and to make the testing more precise another number is used, which is 28. Accordingly the network will be tested with three numbers of neurons; 26, 27 and 28.

Each number will be tested twice, once with a LF and another with a TANH as an activation functions for the output layer. **Table 4.1** illustrates the results of 6 different architectures, the best architecture that produced the highest classification accuracy is (54-26-1) with an LF activation function in the output layer; this architecture represents 54 input variables, one hidden layer of 26 neurons and one output layer, the classification accuracy for this architecture is 70.7%. All the remaining architectures produced low classification accuracy varying in an interval [69.6%- 50.3%].

Table 4. 1: Classification Accuracy with one hidden layer

Network Architecture	Activation Function	Classification Accuracy
54-26-1	TANH	50.3%
54-26-1	LF	70.7%
54-27- 1	TANH	55.2%
54-27- 1	LF	69.6%
54-28-1	TANH	53.2%
54-28-1	LF	64.3%

4.2.2. Models' Building using Two Hidden Layers

For the first hidden layer we obtained three different numbers of neurons; 26, 27 and 28, so by applying step 2 from the iterative process, number of neurons in the second layer will be as follows:

i. When the number of neurons in the first hidden layer was 26, the number of neurons in the second hidden layer will be 26/2 which is 13 neurons. The network architecture will be 54-26-13-1.

ii. When the number of neurons in the first hidden layer was 27, the number of neurons will be 27/2 which is 13.5, since this is not a fix number applying the floor operation will produce 13, applying the ceil operation will produce 14; therefore this will produce two architectures 54-27-13-1, and 54-27-14-1.

iii. When the number of neurons in the first hidden layer was 28, the number of neurons in the second hidden layer will be 28/2 which is 14 neurons. The network architecture will be 54-28-14-1.

Consequently, five different architectures will be tested when the number of the hidden layers is two. As noticed from Table 4.2, the best classification accuracy is 87% when the

network architecture is 54-26-13-1 with LF. The results for the other architectures vary in an interval [54.5%- 86.9%].

Table 4. 2 :Classification Accuracy with two hidden layers

Network Architecture	Activation Function	Classification Accuracy
54-26-13-1	TANH	54.5%
54-26-13-1	LF	87%
54-27-13-1	TANH	58.5%
54-27-13-1	LF	78.7%
54-27-14-1	TANH	60%
54-27-14-1	LF	85%
54-28-14-1	TANH	58.1%
54-28-14-1	LF	86.9%

4.2.3. Models' Building using three Hidden Layers

We obtained four different architectures when the number of hidden layers was two, 54-26-13-1, 54-27-13-1, 54-27-14-1 and 54-28-14-1, when a third hidden layer is added number of neurons in this layer will be as follow:

i. When the network architecture was 54-26-13-1, the number of neurons in the second hidden layer will be 13, therefore number of neurons in the third hidden layer is 13/2=6.5, and since the result is not a fix number applying the floor operation will produce 6, applying the ceil operation will produce 7, then the result is two architectures, 54-26-13-6-1 and 54-26-13-7-1.

ii. When the network architecture was 54-27-13-1, the number of neurons in the second hidden layer will be 13, therefore number of neurons in the third hidden layer is 13/2=6.5, and since the result is not a fix number applying the floor

operation will produce 6, applying the ceil operation will produce 7, then the result is two architectures, 54-27-13-6-1 and 54-27-13-7-1.

iii. When the network architecture was 54-27-14-1, the number of neurons in the second hidden layer will be 14, therefore number of neurons in the third hidden layer is 14/2=7. The network architecture will be in this case 54-27-14-7-1.

iv. When the network architecture was 54-28-14-1, the number of neurons in the second hidden layer will be 14, therefore number of neurons in the third hidden layer is 14/2=7. The network architecture will be in this case 54-28-14-7-1.

Table 4.3 Illustrates the classification accuracy when using three hidden layers; the highest accuracy was achieved with the network architecture of the form 54-28-14-7-1 with LF is 91.8% classification accuracy. The other architectures achieved results varying in an interval [58.1%-91.6%].

Table 4.3: Classification Accuracy with three hidden layers

Network Architecture	Activation Function	Classification Accuracy
54-26-13-6-1	TANH	58.1%
54-26-13-6-1	LF	88.1%
54-26-13-7-1	TANH	59.8%
54-26-13-7-1	LF	88%
54-27-13-6-1	TANH	59.2%
54-27-13-6-1	LF	91.6%
54-27-13-7-1	TANH	59%
54-27-13-7-1	LF	87.2%
54-27-14-7-1	TANH	59.6%
54-27-14-7-1	LF	90.9%
54-28-14-7-1	TANH	59.4%
54-28-14-7-1	LF	91.8%

4.2.4. Models' Building using Four, Five and Six Hidden Layers

Building the network with four hidden layers needs to use the architectures that were obtained using three hidden layers, also to build the network with five hidden layers, the networks that were built with four hidden layers must be used, and finally the highest number of hidden layer that can be achieved is six, by using the obtained networks with five hidden layers.

Table 4.4 illustrates the results obtained using four hidden layers; we can notice that the highest classification accuracy which was 94.5% was obtained with network architecture 54-27-14-7-3-1 the architectures were built with LF. The remaining architectures gave results varying in an interval [51.8%-93%].

Table 4.4: Classification Accuracy with four hidden layers

Network Architecture	Activation Function	Classification Accuracy
54-26-13-6-3-1	TANH	60.3%
54-26-13-6-3-1	LF	88.5%
54-26-13-7-3-1	TANH	60.7%
54-26-13-7-3-1	LF	93%
54-26-13-7-4-1	TANH	59.8%
54-26-13-7-4-1	LF	88.9%
54-27-13-6-3-1	TANH	59%
54-27-13-6-3-1	LF	89.9%
54-27-13-7-3-1	TANH	59.2%
54-27-13-7-3-1	LF	91%
54-27-13-7-4-1	TANH	51.8%
54-27-13-7-4-1	LF	86.3%
54-27-14-7-3-1	TANH	60.5%
54-27-14-7-3-1	LF	94.5%
54-27-14-7-4-1	TANH	59.8%
54-27-14-7-4-1	LF	85.6%
54-28-14-7-3-1	TANH	56.1%
54-28-14-7-3-1	LF	84.9%
54-28-14-7-4-1	TANH	58.9%
54-28-14-7-4-1	LF	85%

Table 4.5 illustrates the results obtained using five hidden layers, we can notice that the highest classification accuracy is 91.8% which was obtained using 54-27-14-7-3-1-1 network architecture with LF. The remaining architectures gave results varying in an interval [49.6%-86.7%].

Table 4.5: Classification Accuracy with five hidden layers

Network Architecture	Activation Function	Classification Accuracy
54-26-13-6-3-1-1	TANH	56.1%
54-26-13-6-3-1-1	LF	73.2%
54-26-13-7-3-1-1	TANH	59.4%
54-26-13-7-3-1-1	LF	85.2%
54-26-13-7-4-2-1	TANH	56.7%
54-26-13-7-4-2-1	LF	84.1%
54-27-13-6-3-1-1	TANH	49.6%
54-27-13-6-3-1-1	LF	74.7%
54-27-13-7-3-1-1	TANH	53.4%
54-27-13-7-3-1-1	LF	74.9%
54-27-13-7-4-2-1	TANH	55%
54-27-13-7-4-2-1	LF	86.5%
54-27-14-7-3-1-1	TANH	59%
54-27-14-7-3-1-1	LF	91.8%
54-27-14-7-4-2-1	TANH	58.3%
54-27-14-7-4-2-1	LF	86.7%
54-28-14-7-3-1-1	TANH	49.6%
54-28-14-7-3-1-1	LF	67.6%
54-28-14-7-4-2-1	TANH	60.7%
54-28-14-7-4-2-1	LF	81.6%

Table 4.6 illustrates the results obtained using six hidden layers, we can notice that the highest classification accuracy is 89% which was obtained using 54-27-13-7-4-2-1-1 network architecture with LF. The remaining architectures gave results varying in an interval [50.5%-82%].

Table 4.6: Classification Accuracy with six hidden layers

Network Architecture	Activation Function	Classification Accuracy
54-26-13-7-4-2-1-1	TANH	50.5%
54-26-13-7-4-2-1-1	LF	77%
54-27-13-7-4-2-1-1	TANH	58.7%
54-27-13-7-4-2-1-1	LF	89%
54-27-14-7-4-2-1-1	TANH	57.8%
54-27-14-7-4-2-1-1	LF	82%
54-28-14-7-4-2-1-1	TANH	54%
54-28-14-7-4-2-1-1	LF	65.8%

4.2.5. Summary of the results

Table 4.7 illustrates the summary of the best results we obtained using different hidden layers for building the neural network that will be used as the final model for the clinical decision support system.

We started building the models with one hidden layer and ended with 6 hidden layers. Where we can notice that the best architecture was obtained by using four hidden layers; the architecture is 54-27-14-7-3-1 and obtained 94.5% classification accuracy. Using one hidden layer did not achieve good results, when another hidden layer has been added the best classification accuracy has risen from 70.7% to 87% and after that rise to 91.8% when add three hidden layer. By adding a four layer the highest accuracy has been obtained with a percentage of 94.5%. Adding five and six hidden layers lowered the accuracy and did not improve the results. These results can be explained as follows: If you have too few hidden layers, you will get high training error and high generalization error due to underfitting, this was noticed in the low accuracy we obtained when the network was built with one, two and three hidden layers. Also, if you have too many hidden layers, you may get low training error but still have high generalization error due to overfitting, as was noticed when we built the networks with 5 and 6 hidden layers.

From this we can conclude that; experiments are the only way to determine the best neural network architecture that is suitable for solving a specific problem.

Table 4.7: Summary of Results Regarding Multiple Hidden Layers

Number of hidden layers	Best Classification Accuracy %	Best Network Architecture
1	70.7%	54-26-1
2	87%	54-26-13-1
3	91.8%	54-28-14-7-1
4	**94.5%**	54-27-14-7-3-1
5	91.8%	54-27-14-7-3-1-1
6	89%	54-27-13-7-4-2-1-1

4.3. Detailed Results and Discussion

The effectiveness of the chosen neural network architecture was evaluated on the gynecological disease test set for each fold independently. After that, the average accuracy is computed to determine how well the chosen architecture will generalize for new samples that it never seen before. The following subsection will introduce the results for each test set of the 10 folds (runs). Where the experimental results of the ten test sets have also been presented as a confusion matrix; a confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such a system is commonly evaluated using the data in the matrix. In a confusion matrix, columns represent the predicted data, while rows represent the actual data (Yan et al., 2006) (Haykin, 2001).

4.3.1. Results of the First Run

Table 4.8 illustrated the confusion matrix of the first run. The accuracy for the Polyps (Pol) and Infection (Inc) diseases is ((4/5)*100%) which is 80%, while the rest diseases it is ((5/5) * 100%) which is 100%. This gives a total accuracy for the first run by applying the equation (4.1) that it is in section 4.1, (A_{f1} = (53/55)* 100%) = 96%. We can notice from the

confusion matrix that 20% from the samples of the Prolapse (Pro) and Infection (Inc) diseases were misclassified.

Table 4.8: Confusion Matrix of the First Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	1	0	0	0	0	0	0	0	0	0	80%
Inc	0	4	0	0	0	1	0	0	0	0	0	80%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												96%

4.3.2. Results of the Second Run

Table 4.9 illustrated the confusion matrix of the two run. The accuracy for the Polyps(Pol), Infection(Inc),Cancer, Amenorrhea (Amen), Abortion(Abor), Dysmenorrhea(Dys) and Infertility(Inf) diseases is ((5/5)*100%) which is 100%, for the Fibroids(Fib) disease it is ((3/5) * 100%) which is 60%, for the Prolapse(Pro), Hyperplasia(Hyper) and Migrants(Mg) diseases are ((4/5) * 100%) which is 80%. This gives a total accuracy for the second run (A_{f2}= (50/55)* 100%) = 91%. We can notice from the confusion matrix that 20% from the samples of the Prolapse (Pro), Hyperplasia (Hyper) and Migrants (Mg) diseases were misclassified, and 40% from the samples of the Fibroids (Fib) were misclassified.

Table 4.9: Confusion Matrix of the Second Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	5	0	0	0	0	0	0	0	0	0	0	100%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	3	1	1	0	0	0	0	0	0	60%
Pro	0	0	0	4	1	0	0	0	0	0	0	80%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	1	0	0	4	0	0	0	0	0	80%
Mg	0	0	0	0	0	1	4	0	0	0	0	80%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												91%

4.3.3. Results of the Third Run

Table 4.10 illustrated the confusion matrix of the third run. The accuracy for the Hyperplasia (Hyper), Migrants(Mg), Amenorrhea(Amen), Infertility(Inf), Dysmenorrhea (Dys) and Prolapse(Pro) diseases is ((5/5)*100%) which is 100%, for the Polyps(Pol), Cancer and Abortion(Abor) diseases it is ((4/5) * 100%) which is 80%, while Infection (Inc) and Fibroids(Fib) diseases it is ((3/5) * 100%) which is 60% . This gives a total accuracy for the third run (A_{f3}= (48/55)* 100%) = 87%. We can notice from the confusion matrix that 20% from the samples of the Polyps (Pol), Cancer and Abortion (Abor) diseases were misclassified, and 40% from the samples of the Infection (Inc) and Fibroids (Fib) were misclassified.

Table 4.10: Confusion Matrix of the Third Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	0	0	0	0	1	0	0	0	0	0	80%
Inc	0	3	0	0	0	2	0	0	0	0	0	60%
Fib	2	0	3	0	0	0	0	0	0	0	0	60%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	4	1	0	0	0	0	0	80%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	1	0	4	0	0	80%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												87%

4.3.4. Results of the Fourth Run

Table 4.11 illustrated the confusion matrix of the fourth run. The accuracy for the Polyps (Pol) disease is ((4/5)*100%) which is 80%, while the rest diseases it is ((5/5) * 100%) which are 100%. This gives a total accuracy for the fourth run (A_{f4} = (54/55)* 100%) = 98%. We can notice from the confusion matrix that 20% from the samples of the Polyps (Pol) disease were misclassified.

Table 4.11: Confusion Matrix of the Fourth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	1	0	0	0	0	0	0	0	0	0	80%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												98%

4.3.5. Results of the Fifth Run

Table 4.12 illustrated the confusion matrix of the fifth run. The accuracy for the Dysmenorrhea (Dys) disease is ((4/5)*100%) which is 80%, while the rest diseases it is ((5/5) * 100%) which is 100%. This gives a total accuracy for the fifth run (A_{f5} = (54/55)* 100%) = 98%. We can notice from the confusion matrix that 20% from the samples of the Dysmenorrhea (Dys) diseases were misclassified.

Table 4.12: Confusion Matrix of the Fifth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	5	0	0	0	0	0	0	0	0	0	0	100%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	1	4	0	80%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												98%

4.3.6. Results of the Sixth Run

Table 4.13 illustrated the confusion matrix of the sixth run. The accuracy for the Polyps (Pol) disease is ((4/5)*100%) which is 80%, while the rest diseases it is ((5/5) * 100%) which is 100%. This gives a total accuracy for the sixth run (A_{f6} = (54/55)* 100%) = 98%. We can notice from the confusion matrix that 20% from the samples of the Polyps (Pol) diseases were misclassified.

Table 4.13: Confusion Matrix of the Sixth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	0	0	1	0	0	0	0	0	0	0	80%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												98%

4.3.7. Results of the Seventh Run

Table 4.14 illustrated the confusion matrix of the seventh run. The accuracy for the Cancer, Hyperplasia (Hyper), Migrants (Mg), Amenorrhea (Amen) and Dysmenorrhea (Dys) diseases is ((5/5)*100%) which is 100%, for the Infertility(Inf), Prolapse (Pro), Fibroids (Fib) and Polyps(Pol) diseases it is ((4/5) * 100%) which is 80%, while Infection (Inc) and Abortion(Abor) diseases it is ((2/5) * 100%) which is 40% . This gives a total accuracy for the seventh run (A_{f7}= (45/55)* 100%) = 82%. We can notice from the confusion matrix that 20% from the samples of the Infertility (Inf), Prolapse (Pro), Fibroids (Fib) and Polyps (Pol) diseases were misclassified, and 60% from the samples of Infection (Inc) and Abortion (Abor) were misclassified.

Table 4.14: Confusion Matrix of the Seventh Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	0	0	0	1	0	0	0	0	0	0	80%
Inc	0	2	1	0	0	0	2	0	0	0	0	40%
Fib	0	1	4	0	0	0	0	0	0	0	0	80%
Pro	0	0	0	4	1	0	0	0	0	0	0	80%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	1	2	2	0	0	40%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	1	4	80%
												82%

4.3.8. Results of the Eighth Run

Table 4.15 illustrated the confusion matrix of the eighth run. The accuracy for the Prolapse (Pro) and Abortion (Abor) diseases is ((4/5)*100%) which is 80%, while for the rest diseases it is ((5/5) * 100%) which is 100%. This gives a total accuracy for the eighth run (A_{f8} = (53/55)* 100%) = 96%. We can notice from the confusion matrix that 20% from the samples of the Prolapse (Pro) and Abortion (Abor) diseases were misclassified.

Table 4.15: Confusion Matrix of the Eighth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	5	0	0	0	0	0	0	0	0	0	0	100%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	4	1	0	0	0	0	0	0	80%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	4	1	0	80%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												96%

4.3.9. Results of the Ninth Run

Table 4.16 illustrated the confusion matrix of the ninth run. The accuracy for the all diseases are ((5/5)*100%) which is 100%. This gives a total accuracy for the ninth run (A_{f9}= (55/55)* 100%) = 100%, in this fold no sample has been misclassified.

Table 4.16: Confusion Matrix of the Ninth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	5	0	0	0	0	0	0	0	0	0	0	100%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												100%

4.3.10. Results of the Tenth Run

Table 4.17 illustrated the confusion matrix of the tenth run. The accuracy for the Polyps (Pol) disease is ((4/5)*100%) which is 80%, while the rest diseases they are ((5/5) * 100%) which is 100%. This gives a total accuracy for the tenth run (A_{f10} = (54/55)* 100%) = 98%. We can notice from the confusion matrix that 20% from the samples of the Polyps (Pol) diseases were misclassified.

Table 4.17: Confusion Matrix of the Tenth Run

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Accuracy
Pol	4	0	0	0	1	0	0	0	0	0	0	80%
Inc	0	5	0	0	0	0	0	0	0	0	0	100%
Fib	0	0	5	0	0	0	0	0	0	0	0	100%
Pro	0	0	0	5	0	0	0	0	0	0	0	100%
Cancer	0	0	0	0	5	0	0	0	0	0	0	100%
Hyper	0	0	0	0	0	5	0	0	0	0	0	100%
Mg	0	0	0	0	0	0	5	0	0	0	0	100%
Amen	0	0	0	0	0	0	0	5	0	0	0	100%
Abor	0	0	0	0	0	0	0	0	5	0	0	100%
Dys	0	0	0	0	0	0	0	0	0	5	0	100%
Inf	0	0	0	0	0	0	0	0	0	0	5	100%
												98%

4.3.11. Summary of the Results

To summarize the results of the testing results of the ten folds, **Table 4.18** illustrates the classification accuracy for all results. As we notice the highest classification accuracy obtained in fold 9 with accuracy of 100% and the lowest accuracy obtained was in fold 7 with accuracy of 82%. Also the total classification accuracy was 94.5%.These results were affected by the nature of the data splits for each fold; the one fold that obtained 100% accuracy may had examples in the training sets that were very similar to the data pattern that was in the test set, so this produced high classification accuracy. But in the other fold that obtained 82% accuracy the examples that the network was trained with were not close to the data pattern in the test set, which produced low classification accuracy.

Table 4.18: Summary of All Classification Accuracy for All Folds

Test Sets	Classification Accuracy %
Fold 1	98%
Fold 2	91%
Fold 3	87%
Fold 4	98%
Fold 5	98%
Fold 6	98%
Fold 7	82%
Fold 8	96%
Fold 9	100%
Fold 10	98%
Average Accuracy	**94.5%**

Table 4.19 illustrated the experimental results of the 10 test sets have also been presented as a confusion matrix.

Table 4.19: The Confusion Matrix of the diagnostic result of the gynecological diseases: 10- fold cross validation

Disease Type	Pol	Inc	Fib	Pro	Cancer	Hyper	Mg	Amen	Abor	Dys	Inf	Row sum
Pol	44	2	0	1	2	1	0	0	0	0	0	50
Inc	0	44	1	0	0	3	2	0	0	0	0	50
Fib	2	1	45	1	1	0	0	0	0	0	0	50
Pro	0	0	0	47	3	0	0	0	0	0	0	50
Cancer	0	0	0	0	49	1	0	0	0	0	0	50
Hyper	0	0	1	0	0	49	0	0	0	0	0	50
Mg	0	0	0	0	0	1	49	0	0	0	0	50
Amen	0	0	0	0	0	0	0	50	0	0	0	50
Abor	0	0	0	0	0	0	2	2	45	1	0	50
Dys	0	0	0	0	0	0	0	0	1	49	0	50
Inf	0	0	0	0	0	0	0	0	0	1	49	50
column sum	46	47	47	49	55	55	53	52	46	51	49	550

Form the confusion matrix shown in **Table 4.19** the accuracy for each class can be calculated as presented in **Table 4.20**, where the proportion of the disease type in the test dataset that have been correctly recognized by ANN is shown .We notice that the Polyps

and Infection diseases got the lowest classification accuracy with a percentage of 88%; the highest classification accuracy was for the Amenorrhea disease with a percentage of 100%.

Table 4.20: Proportion of correctly classified diseases

Disease Type	Accuracy
Polyps	88%
Infection	88%
Fibroids	90%
Prolapse	94%
Cancer	98%
Hyperplasia	98%
Migrants	98%
Amenorrhea	100%
Abortion	90%
Dysmenorrhea	98%
Infertility	98%
	94.5%

Chapter Five

5. Conclusion and Future Work

5.1. Conclusions and Future Work

The number of deaths caused by medical errors has been estimated to be 98,000 individuals per year (Kiyan and Yildirim, 2003). Therefore, an accurate diagnosis will play a major role in the life's saving of a patient. Beside that many researchers developed clinical decision support system to assist physicians in the diagnosis process for multiple diseases, where these systems proofed its effectiveness in obtaining high diagnosis accuracy (Kayaer and Yildirim, 2003) (Kiyan and Yildirim, 2003).

Based on this, this thesis proposed a CDSS for gynecological disease diagnosis. The gynecological diseases were Polyps, Infection, Fibroids, Prolapse, Cancer, Endometrial hyperplasia, Migrants, Amenorrhea, Abortion, Dysmenorrhea and Infertility. The proposed system was developed using one of the most widespread machine learning techniques; a MLP feed forward neural network and trained using the Rprop training algorithm. It consisted of: an input layer with 54 neurons (representing the input variables of each patient such as, age, marital status, pregnant, etc.), 4 hidden layers (where the number of neurons in the first, second, third and fourth hidden layer were 27, 14, 7 and 3 respectively) and an output layer (that produced the type of the disease the patient suffered from). Moreover, a ten-fold cross validation was used to access the generalization of the proposed system using 550 patients' medical recodes collected from the gynecology clinics at JUH. An accuracy of 88%, 88%, 90%, 94%, 98%, 98%, 98%, 100%, 90%, 98% and 98% were obtained for the Polyps, Infection, Fibroids, Prolapse, Cancer, Endometrial hyperplasia, Migrants,

Amenorrhea, Abortion, Dysmenorrhea and Infertility respectively, giving an average of 94.4% classification accuracy.

From these results we conclude that; the proposed system proofed its usefulness in the support of clinic diagnosis decision of gynaecological diseases. Moreover, if more dataset was used for training the proposed system, that would give more robust results. Moreover, some limitations were noticed in this thesis such as: some medical records contained missing values because they were filled with resident doctors, which made it impossible to use these records in the gynaecological disease dataset.

In future, this work can be extended by using other types of ANNs such as, GRNN, PNN and RBF to compare the results obtained from them with the one obtained using the MLP. Also, we can apply one of the feature selection algorithms on the gynaecological disease dataset in order to choose the medical variables that play a major role in the diagnosis process which may give more accurate results. Moreover, applying other machine learning methods such as Artificial Immune Systems, Neuro-Fuzzy and Decision Trees and comparing the results obtained with the one obtained using the ANN. Finally, since this system was built using the medical records that were collected from JUH which is a training hospital, we can apply the system on the medical records collected from other types of hospitals such as military, public and private hospitals in order to compare the results obtained with the one obtained using JUH's medical records.

REFRENCES

AlTimemy.A. and Al Naima.F.(2010), Comparison of Different Neural Network Approaches for the Prediction of Kidney Dysfunction, **International Journal of Biological and Life Sciences,** 6 (2), 84-90.

Bascil.S.,and Temurtas.F.(2009), A Study on Hepatitis Disease Diagnosis Using Multilayer Neural Network with Levenberg Marquardt Training Algorithm, **Journal of Medical Systems,** 33 (1), 1-4

Brause Rüdiger W. (2001), Medical Analysis and Diagnosis by Neural Networks, **In Proceedings of the Second International Symposium on Medical Data Analysis** London, UK, 1-13.

Chard, T, Schtriner, A (1990**), Expert systems in obstetrics and gynaecology**, Bailliere's Clinical Obstetrics and Gynaecology, 4(4), 815-840.

Duda, R., Hart, P. and Stork, D. (2001), **Pattern Classification**, (2nd ed) , NewYork: John Wiley & Sons, Inc.

Edozien C.Leroy (2007), Risk management in gynaecology: Principles and Practice, **Best Practice and Research Clinical Obstetrics and Gynaecology**, 2 (4) , 713-725.

Fausett, L Laurene (2001),**Fundamentals of Neural Networks**, **Architectures, Algorithms and Application**,(1st ed), New Jersey: Pearson Prentice Hall Education , Inc.

Freeman , J and Skapura , D ,(1991) , **Neural Networks Algorithms, Applications, and Programming Techniques,** (1st ed), New York: Addison-Wesley Publishing Company, Inc.

Geoffrey, H. and Terrence, S. (1999) **Unsupervised Learning: Foundations of Neural Computation** (1st ed), Massachusetts: MIT Press**.**

Ghahramani, Saeed (2000). **Fundamentals of Probability,** (2nd ed). New Jersey: Prentice Hall

Hagan, M, Dcmuth, H, Beale, M.(1996), **Neural Network Design,** (1st ed), USA: PWS Publishing Company.

Haykin, Simon. (2001), **Neural Networks a Comprehensive Foundation,** (2nd ed), India: Pearson Prentice Hall Education , Inc.

Kayaer.K. and Yildirim.T.(2003), Medical Diagnosis on Pima Indian Diabetes Using General Regression Neural Networks**, In Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing,** Istanbul, turkey,181-184 **.**

Kiyan, T. and Yildirim, T. (2003), Breast Cancer Diagnosis Using Statistical Neural Networks, **In International XII Turkish Symposium on Artificial Intelligence and Neural Networks**, Tainn, Turkey, 754-760.

Kong, G., Xu, D. and Yang, J. (2008), Clinical Decision Support Systems: A Review on Knowledge Representation and Inference Under Uncertainties, **International Journal of Computational Intelligence Systems,** 1(2), 159-167.

Krose, B and Smagt, P. (1996), **An introduction To Neural Networks**, (8th ed), Amsterdam: The University of Amsterdam.

Machado, Lucila Ohno.(1996),**Medical Application of Artificial Neural Networks: Connectionist Models of Survival**, Master's thesis, Department of Computer Science, University of Stanford, USA.

Mangalampalli , A, Mangalampalli , S, Chakravarthy , R, Jain , A. (2006) , A neural network based clinical decision-support system for efficient diagnosis and fuzzy-based prescription of gynecological diseases using homoeopathic medicinal system , **Expert Systems with Applications** , 30(1) , 109-116

Mehrotra, K , Chilukuri , K, Mohan and Ranka, S .(1996) ,**Elements of Artificial Neural Networks**,(1st ed), New York: Addison-Wesley Publishing Company, Inc.

Menéndez, L, Juez , F, Lasheras, F, Riesgo ,J.(2010), Artificial neural networks applied to cancer detection in a breast screening programme , **Mathematical and Computer Modelling**, 52(7), 983-991.

Miller R. and Geissbuhler A. (2007), Diagnostic Decision Support Systems In: Berner E. (Ed), **Clinical Decision Support Systems,** (2nd ed), (pp 99-125), New-York:Springer.

Mitchell, Tom (1997), **MachineLearning**, (1st ed), McGraw-Hill Science.

Musen Mark A. (1997), Patient Record. In: J.van Bemmel and Mark A. Musen (Ed), **Handbook of medical informatics**, (pp 99 115), Houten: Springer.

Paulin, F. and Santhakumaran, A. (2011), Classification of Breast Cancer by Comparing Backpropagation Training Algorithms, **International Journal on Computer Science and Engineering**, 3(1), 327-332.

Penney, G and Foy, R (2007), Do clinical guidelines enhance safe practice in obstetrics and gynaecology? , **Best Practice and Research Clinical Obstetrics and Gynaecology**, 21(4), 657-673.

Ramalingam, V., Palaniappan, B., Panchanatham, N. and Palanivel, S. (2006), Measuring Advertisement Effectiveness – A Neural Network Approach, **Expert Systems with Applications,** 31(1), 159–163.

Rasheed, Zeehasham (2009), **Adaptive Fuzzy Logic Based Framework for Handling Imprecision and Uncertainty in Pattern Classification of Bioinformatics Datasets**, Master's thesis, Department of Computer Science, King Fahd University of Petroleum and Minerals, Saudi Arabia.

Riedmiller, M. and Braun ,H. (1993), A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, **Neural Networks, IEEE International Conference, San Francisco, CA , USA** ,586-591.

Szpurek, D, Moszynski, R, Smolen, A, Sajdak , S. (2005), Artificial neural network computer prediction of ovarian malignancy in women with adnexal masses , **International Journal of Gynecology & Obstetrics**, 89(2), 108-113.

Temurtas.H., Yumusak.N., Temurtas.F.(2009),A Comparative Study on Diabetes Disease Diagnosis Using Neural Networks, **Expert Systems with Applications,** 36(4), 8610-8615.

Thammano, A., Moolwong, J.(2009),A New Computational Intelligence Technique Based on Human Group Formation, **Expert Systems with Applications**, 37(2), 1628–1634

Thornett, Andrew M. (2001), Computer Decision Support Systems in General Practice, **International Journal of Information Management,** 21(1), 39-47.

Vahdani.P., Alavian.S., Aminzadeh.Z., Raoufy.M., Gharibzadeh.S., Vahdani.G., Fekri.S., and Eftekhari.P.(2009), Using Artificial Neural Network to Predict Cirrhosis in Patients with Chronic Hepatitis B Infection with Seven Routine Laboratory Findings, **Hepatitis Monthly ,** 9(4), 271-275

Veropoulos, Konstantinos (2001), **Machine Learning Approaches to Medical decision Making,** Doctoral thesis, Department of Engineering Mathematics, University of Bristol, United Kingdom.

Wilding, P, Morgan, M, Grygotis, A, Shoffner A , Rosato E (1994),Application of backpropagation neural networks to diagnosis of breast and ovarian cancer, **Cancer Letters**, 77(2), 145-153.

Xu,S. and Chen, L. (2008), A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining**, In the 5th International Conference on Information Technology and Applications,** Cairns, Australia, 683-686

Yan, H., Jiang, Y., Zheng, J., Peng, C. and Li, Q. (2006), A Multilayer Perceptron-Based Medical Decision Support System for Heart Disease Diagnosis, **Expert Systems with Applications**, 30(2), 272-281.

تشخيص أمراض الطب النسائي بإستخدام نظام دعم القرارات السريرية

إعداد
إلهام فهمي أحمد

المشرف
الدكتور محمد الشريدة

المشرف المشارك
الدكتور كميل فرام

ملخص

إن تشخيص أمراض النساء هي واحدة من القضايا الهامة في جميع أنحاء العالم وكذلك في الأردن. والتى تعتمد أساسا على تناول الشكوى المرضية يتبعها تحليل وربط الأعراض بسبب كثرتها وتنوعها وتشابها ، والطبيب النسائي قد يغيب عن باله بلحظه ما من غير قصد بعض الأعراض ، وبالتالي فهناك حاجة لوضع نظام لدعم اتخاذ القرارات من أجل المساعدة في تشخيص أمراض النساء .

قي العقود الأخيرة ، اتجه الكثير من الباحثين إلى تطوير أنظمة لدعم القرارات الطبية، حيث أن الهدف من هذه الأنظمة مساعدة الطبيب في عملية تشخيص الأمراض وبالتالي التقليل من نسبة الخطأ الطبي الممكن حدوثه في عملية التشخيص. بناءا على ذلك تم اعتماد الكثير من المنهجيات والوسائل لتحقيق هذا الهدف. أحد أشهر هذه الوسائل والتي تنتمي لعائلة الحسابات الذكية هي: الشبكات العصبية الاصطناعية المستوحاة فكرتها من الأنظمة العصبية البيولوجية.

في هذه الرسالة، تم تصميم نظاما لدعم القرار الطبي لتشخيص أحدى عشر مرضا من الأمراض النسائية بواسطة شبكة عصبية اصطناعية متعددة الطبقات ذات تغذية أمامية. تستقبل طبقة الإدخال 54 متغير مدرجين ضمن إحدى عشر مجموعة وتم ترميزهم بالطريقة المقترحة بهذه الرسالة. في حين أن طبقة الإخراج ستنتج نوع المرض الذي تعاني منه المريضة.لقد تم استعمال طريقة تكرارية لتحديد عدد الطبقات المخفية و عدد الخلايا العصبية في هذه الطبقات. لتدريب الشبكة تم استعمال خوارزمية التدريب (Rprop). بالإضافة إلى ذلك تم تطبيق طريقة المجموعات العشر للتحقق من مقدرة الشبكة العصبية على التعامل مع بيانات جديدة لم تتعامل معها في مرحلة التدريب. لقد تم تدريب وإجراء الاختبارات على النظام بواسطة 550 سجل طبي تم الحصول عليها من عيادات الأمراض النسائية في مستشفى الجامعة الأردنية.و لقد أظهرت التجارب بأن النظام المقترح في هذه الرسالة استطاع تشخيص الإحدى عشر مرض من الأمراض النسائية بدقة 94.5 %.